Comparison between adaptive and uniform discontinuous Galerkin simulations in dry 2D bubble experiments

Andreas Müller[a,*], Jörn Behrens[b], Francis X. Giraldo[a], Volkmar Wirth[c]

[a] *Department of Applied Mathematics, Naval Postgraduate School, Monterey, CA 93943-5216 (USA)*
[b] *KlimaCampus, University of Hamburg, Am Grindelberg 5, D-20144 Hamburg (Germany)*
[c] *Institute for Atmospheric Physics, University of Mainz, Becherweg 21, D-55127 Mainz (Germany)*

Abstract

Adaptive mesh refinement generally aims to increase computational efficiency without compromising the accuracy of the numerical solution. However it is an open question in which regions the spatial resolution can actually be coarsened without affecting the accuracy of the result. This question is investigated for a specific example of dry atmospheric convection, namely the simulation of warm air bubbles. For this purpose a novel numerical model is developed that is tailored towards this specific application. The compressible Euler equations are solved with a Discontinuous Galerkin method. Time integration is done with an IMEX-method and the dynamic grid adaptivity uses space filling curves via the AMATOS function library. So far the model is able to simulate dry flow in two-dimensional geometry without subgrid-scale modeling. The model is tested with three standard test cases.

An error indicator is introduced for a warm air bubble test case which allows one to compare the accuracy between different choices of refinement regions without knowing the exact solution. Essentially this is done by comparing features of the solution that are strongly sensitive to spatial resolution. For the rising warm air bubble the additional error by using adaptivity is smaller than 1% of the total numerical error if the average number of elements used for the adaptive simulation is about a factor of two times smaller than the number used for the simulation with the uniform fine-resolution grid. Correspondingly the adaptive simulation is almost two times faster than the uniform simulation. Furthermore the adaptive simulation is more accurate than a uniform simulation when both use the same CPU-time.

Keywords: Adaptive Mesh Refinement, Discontinuous Galerkin, IMEX, Meteorology, Dry Warm Air Bubble

1. Introduction

Significant progress in numerous areas of scientific computing comes from the steadily increasing capacity of computers and the advances in numerical methods. An example is the simulation of the Earth's atmosphere, which has proven to be extremely challenging owing to its multiscale and multi-process nature. Even with today's computers it is impossible to explicitly represent all scales and all processes involved. To overcome this difficulty one resorts to empirically-based closure approaches – called "parameterisations" – that try to capture the unresolved aspects of the problem. This introduces errors.

One possibility for reducing the computational effort of simulations is given by adaptive mesh refinement. It allows the adaptation of the spatial and temporal resolution to local properties of the atmosphere. This adaptation is controlled by refinement criteria. Adaptive mesh refinement has been applied successfully to atmospheric sciences for over 20 years [1, 2]. Recently, this technique has found increased interest, since new

[*] Corresponding author. Tel.: +1-831-656-2293, fax: +1-831-656-2355.
Email addresses: amueller@anmr.de (Andreas Müller), joern.behrens@zmaw.de (Jörn Behrens), fxgirald@nps.edu (Francis X. Giraldo), vwirth@uni-mainz.de (Volkmar Wirth)

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE
08 NOV 2012

2. REPORT TYPE

3. DATES COVERED
00-00-2012 to 00-00-2012

4. TITLE AND SUBTITLE
Comparison between adaptive and uniform discontinuous Galerkin simulations in dry 2D bubble experiments

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School (NPS),Department of Applied Mathematics,Monterey,CA,93943

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Adaptive mesh refinement generally aims to increase computational efficiency without compromising the accuracy of the numerical solution. However it is an open question in which regions the spatial resolution can actually be coarsened without affecting the accuracy of the result. This question is investigated for a specific example of dry atmospheric convection, namely the simulation of warm air bubbles. For this purpose a novel numerical model is developed that is tailored towards this specific application. The compressible Euler equations are solved with a Discontinuous Galerkin method. Time integration is done with an IMEXmethod and the dynamic grid adaptivity uses space filling curves via the AMATOS function library. So far the model is able to simulate dry flow in two-dimensional geometry without subgrid-scale modeling. The model is tested with three standard test cases. An error indicator is introduced for a warm air bubble test case which allows one to compare the accuracy between different choices of refinement regions without knowing the exact solution. Essentially this is done by comparing features of the solution that are strongly sensitive to spatial resolution. For the rising warm air bubble the additional error by using adaptivity is smaller than 1% of the total numerical error if the average number of elements used for the adaptive simulation is about a factor of two times smaller than the number used for the simulation with the uniform fine-resolution grid. Correspondingly the adaptive simulation is almost two times faster than the uniform simulation. Furthermore the adaptive simulation is more accurate than a uniform simulation when both use the same CPU-time.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**31**	

grid-independent numerical methods of Galerkin type and finite volumes have emerged [3–6]. However there are still many unsolved questions [7]. A more comprehensive exposition of adaptive methods in atmospheric modeling is given in [8].

One important unsolved question is: in which regions can the spatial resolution be coarsened without affecting the accuracy of the simulation? Starting with a uniform simulation the additional error introduced by coarsening the mesh in some regions for an adaptive simulation should be much smaller compared to the inherent numerical error of the uniform simulation. However, for realistic meteorological applications the exact solution is not known and the exact error cannot be computed. Therefore we have to find some approximate measure for the error. It is difficult to solve this task in general. For this reason we focus on a specific meteorological application.

Our final goal is to develop a simulation that is able to cover a cumulus cloud as a whole and resolve smaller eddies at the cloud-environment interface simultaneously. A simulation of this problem using a uniform grid is still far beyond the capacity of today's computing power. However it appears possible when using adaptive mesh refinement. This application is important for meteorological research because the impact of evaporative cooling on the evolution of cumulus clouds is still not fully understood [9]. Furthermore this work should allow important insight into the simulation of scales between mesoscale models on larger domains and large-eddy simulations for smaller scales (sometimes called "terra incognita" [10]).

We developed a novel numerical model that is tailored towards this specific meteorological problem. The compressible Euler equations are solved with a Discontinuous Galerkin method. Time integration is done with an IMEX-method and the dynamic grid adaptivity uses space filling curves via the AMATOS [11] function library. So far we are able to simulate dry flow in two-dimensional geometry without subgrid-scale modeling. This numerical model allows us to compare different adaptive and uniform simulations in a simplified test environment.

The paper is organized as follows. First, in section 2 we give an introduction into the meteorological problem that motivates our work. In section 3 we then describe the numerical methods that we are using. In section 4 we apply our code to three test cases from the literature. Section 5 presents the comparison between adaptive and uniform simulations. The paper ends with a summary and outlook in section 6.

2. Meteorological motivation

A single cumulus cloud can be considered as a prototypical basic element of atmospheric moist convection. The cloud rises through the environmental air owing to its positive buoyancy (fig. 1). Upward motion of the cloud (thick blue arrow) is associated with downward motion in a thin shell surrounding the rising cloud (thin blue arrows) [9]. The induced wind shear at the cloud-environment interface and the different densities of cloudy and environmental air lead to instabilities which eventually result in turbulence [12–14]. The ensuing mixing between moist cloudy and dry environmental air leads to evaporation of cloud droplets. This cools the parcel resulting in negative buoyancy corresponding to a downward force (red arrows). This process is called "buoyancy reversal" [15–17].

Early indications for the significance of buoyancy reversal for cloud dynamics stem from the laboratory experiments of Johari [18]. Johari found that, depending on the strength of the buoyancy reversal, the morphology of the cloud development could be vastly different. Similar results were found in highly idealized numerical two-fluid experiments by Grabowski [19].

These preliminary investigations suggest that buoyancy reversal has an important impact on cloud dynamics. However, owing to their idealized nature it is not possible to draw any firm conclusion about real clouds. On the other hand, numerical weather prediction models and even so-called "cloud resolving models" are not able to explicitly simulate the processes relevant for buoyancy reversal due to their coarse spatial resolution [20]. It is here that we want to take a step forward by developing a new numerical model that is specifically designed to deal with the mixing processes at the cloud boundary.

A Direct Numerical Simulation (DNS) would require a resolution of about 0.1 mm in each direction in order to properly resolve all dynamical scales [20]. In three dimensions and a domain size of 10 km in each spatial direction this amounts to some 10^{24} grid points, which is beyond the capacity of today's computing

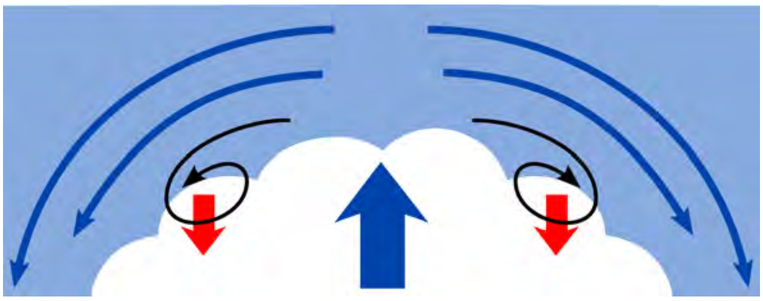

Figure 1: Illustration of buoyancy reversal. The blue arrows demonstrate the mean flow of a rising cloud, the black arrows represent turbulence produced by Kelvin-Helmholtz instability and the red arrows illustrate buoyancy reversal. For further explanation we refer to the text.

power. We, therefore, intend to use Large Eddy Simulation (LES) in combination with an IMEX-method for time integration. The use of an adaptive technique will offer a significant reduction in numerical expense, as it allows us to focus attention to the cloud-environment interface, which is the region where mixing and buoyancy reversal takes place.

Measurements in real clouds have shown a large variety of behavior [21]: there are clouds with steep gradients in the interior (see for example the liquid water content in figure 15 of that reference [21]). On the other hand, smaller clouds often have a fairly smooth interior with discontinuities mostly at the boundary of the cloud (fig. 13 of Damiani et al. [21]). It is the latter which we intend to simulate.

As a first step we consider a 2D dry flow without subgrid-scale modeling. This excludes the processes of buoyancy reversal but allows us to study the adaptive simulation of mixing processes in an idealized framework.

3. Numerical model

As described in the previous section our numerical model is tailored towards the simulation of cumulus clouds which have discontinuities at their boundaries. For this application a Discontinuous Galerkin (DG) discretization in combination with an IMEX-method for time integration should be an excellent choice; the reason is the high-order accuracy and robustness in handling discontinuities of the DG method as well as the large time-steps allowed by the IMEX-method. For reducing the Gibbs phenomenon an artificial viscosity is used.

In the current work the fully compressible Euler equations are used (see equation set 2 in Giraldo and Restelli [22]):

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{U} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{U} \otimes \boldsymbol{U}}{\rho} + p\, \mathbf{I}_2 \right) = -\, \rho\, g\, \hat{\mathbf{k}}, \tag{2}$$

$$\frac{\partial \Theta}{\partial t} + \boldsymbol{\nabla} \cdot \left(\frac{\Theta\, \boldsymbol{U}}{\rho} \right) = 0, \tag{3}$$

with the variables $\left(\rho, \boldsymbol{U}^{\mathrm{T}}, \Theta\right)^{\mathrm{T}}$ where the superscript T stands for transpose, ρ is the density, $\boldsymbol{U} = (\rho\, u, \rho\, w)^{\mathrm{T}}$ is the momentum field, u is the horizontal wind speed, w is the vertical wind speed and $\Theta = \rho\, \theta$ is the density potential temperature. Furthermore we denote the gravitation with g, the divergence operator with $\nabla \cdot$, the tensor product by $\otimes$, the identity matrix in $\mathbb{R}^2$ by $\mathbf{I}_2$ and the unit vector in the vertical direction with $\hat{\mathbf{k}}$.

Pressure p in eq. (2) is given by the equation of state:

$$p = p_0 \left(\frac{R\, \Theta}{p_0} \right)^{\frac{c_p}{c_v}}, \tag{4}$$

with the pressure at height $z = 0\,\mathrm{m}$ given by $p_0 = 10^5\,\mathrm{hPa}$, the gas constant $R = c_p - c_v$ and the specific heats for constant pressure and volume, c_p and c_v. Potential temperature θ is defined by

$$\theta = T \left(\frac{p_0}{p}\right)^{\frac{R}{c_p}} \tag{5}$$

with temperature T. The physical importance of potential temperature is due to its relation with entropy, where the logarithm of potential temperature is proportional to the entropy. In our model we use potential temperature as a variable because this simplifies the extension to moist air in future research.

Atmospheric flow is often approximately in hydrostatic balance, which is defined by

$$\frac{\partial p}{\partial z} = -\rho\, g. \tag{6}$$

This balance can produce numerical instabilities because the remaining terms in the vertical component of eq. (2) are much smaller than the two terms of the hydrostatic balance (6). To avoid this instability we introduce the mean states $\bar{p}$, $\bar{\rho}$ and $\bar{\Theta}$ that are in hydrostatic balance. The mean state of pressure p is defined by $\bar{p} = p(\bar{\Theta})$. These mean states are independent of time t and horizontal position x. The deviation of the variables from the mean state is denoted by $\rho' = \rho - \bar{\rho}$, $\Theta' = \Theta - \bar{\Theta}$ and $p' = p - \bar{p}$. These deviations do not have to be small for all times and everywhere in the domain. The physical variables and also the deviations can vary in time and space. Therefore this splitting into the mean state and the deviation does not restrict the possible applications of our numerical model. But for the accuracy and stability of the simulation it is an advantage to choose the mean state in such a way that the deviation remains as small as possible. By this procedure the set of equations (1) – (3) can be written as

$$\frac{\partial \rho'}{\partial t} + \boldsymbol{\nabla} \cdot \boldsymbol{U} = 0, \tag{7}$$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{U} \otimes \boldsymbol{U}}{\rho} + p'\,\mathbf{I}_2\right) = -\,\rho'\, g\, \hat{\mathbf{k}}, \tag{8}$$

$$\frac{\partial \Theta'}{\partial t} + \boldsymbol{\nabla} \cdot \left(\frac{\Theta\, \boldsymbol{U}}{\rho}\right) = 0. \tag{9}$$

To discretize these equations in space we introduce the commonly used notation

$$\frac{\partial \boldsymbol{q}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{F}\left(\boldsymbol{q}\right) = \boldsymbol{S}\left(\boldsymbol{q}\right), \tag{10}$$

with the vector $\boldsymbol{q} = \left(\rho, \boldsymbol{U}^{\mathrm{T}}, \Theta\right)^{\mathrm{T}}$, the source function

$$\boldsymbol{S}\left(\boldsymbol{q}\right) = \begin{pmatrix} 0 \\ -\rho'\, g\, \hat{\mathbf{k}} \\ 0 \end{pmatrix}, \tag{11}$$

and the flux tensor

$$\mathbf{F}\left(\boldsymbol{q}\right) = \begin{pmatrix} \boldsymbol{U} \\ \boldsymbol{U} \otimes \boldsymbol{U}/\rho + p'\,\mathbf{I}_2 \\ \Theta\, \boldsymbol{U}/\rho \end{pmatrix}. \tag{12}$$

Equation (10) is discretized using the discontinuous Galerkin method which we describe in the following subsection.

3.1. Discontinuous Galerkin Method

In our work we use a nodal discontinuous Galerkin method based on the strong formulation using the Rusanov flux at the cell interfaces. Furthermore, we consider a two dimensional triangular mesh; the extension to a full three dimensional method will remain a task for the future but we envision using tetrahedra or hexahedra for this task. The triangular discontinuous Galerkin method used in our work is described by Giraldo and Warburton [23] for the case of the shallow water equations. Despite a different definition of conserved variables $\boldsymbol{q}$, flux tensor $\mathbf{F}(\boldsymbol{q})$ and source function $\boldsymbol{S}(\boldsymbol{q})$, eq. (10) remains unchanged. Therefore, we repeat in this paper only the main ideas of the discretization.

We start by multiplying eq. (10) with a test function ψ, integrating over an arbitrary element Ω_e and bringing the spatial derivative in front of the test function with integration by parts. Replacing the flux in the boundary terms by a numerical flux $\mathbf{F}^*$ leads to the following equation for the numerical solution $\boldsymbol{q}_N$:

$$\int_{\Omega_e} \left(\frac{\partial \boldsymbol{q}_N}{\partial t} - \mathbf{F}_N \cdot \boldsymbol{\nabla} - \boldsymbol{S}_N \right) \psi(\boldsymbol{x})\, \mathrm{d}\Omega = - \int_{\Gamma_e} \psi(\boldsymbol{x})\, \hat{\mathbf{n}} \cdot \mathbf{F}_N^* \mathrm{d}\Gamma, \tag{13}$$

where Γ_e is the boundary of element Ω_e, $\hat{\mathbf{n}}$ is the outward pointing unit normal vector on Γ_e, $\mathbf{F}_N = \mathbf{F}(\boldsymbol{q}_N)$, $\boldsymbol{S}_N = \boldsymbol{S}(\boldsymbol{q}_N)$, $\mathrm{d}\Omega$ is the area element and $\mathrm{d}\Gamma$ is the line element. Applying again integration by parts gives the *strong formulation* [24]

$$\int_{\Omega_e} \left(\frac{\partial \boldsymbol{q}_N}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{F}_N - \boldsymbol{S}_N \right) \psi(\boldsymbol{x})\, \mathrm{d}\Omega = \int_{\Gamma_e} \psi(\boldsymbol{x})\, \hat{\mathbf{n}} \cdot (\mathbf{F}_N - \mathbf{F}_N^*)\, \mathrm{d}\Gamma. \tag{14}$$

We construct an Nth degree approximation of the solution $\boldsymbol{q}_N(\boldsymbol{x})$ as follows

$$\boldsymbol{q}_N(\boldsymbol{x}) = \sum_{j=1}^{M_N} \psi_j(\boldsymbol{x})\, \boldsymbol{q}_j \tag{15}$$

where $\psi(\boldsymbol{x})$ are the basis functions, $\boldsymbol{q}_j$ is the solution at the $j = 1, \ldots, M_N$ gridpoints of each element where $M_N = \frac{1}{2}(N+1)(N+2)$. As in [23] we use Lagrange polynomials for the basis functions ψ with Fekete points [25] for the interpolation points and Gauss points [26] for the integration. With this combination of interpolation and quadrature points the model can use up to polynomial degree 15; however in this paper we always used polynomial degree 3.

As a measure of the spatial resolution we use the average distance between neighboring Fekete points. We call this value the effective resolution Δx_{eff}. It should be a good measure for the smallest scale that can be present in our numerical model. However this does not imply that other numerical methods use exactly the same number of unknowns. As discontinuities between elements are allowed for DG multiple values occur at the interfaces between elements and increase the number of unknowns. Using these basis functions we get

$$\frac{\partial \boldsymbol{q}_i}{\partial t} = - \int_{\Omega_e} \hat{\psi}_i (\boldsymbol{\nabla} \cdot \mathbf{F}_N - \boldsymbol{S}_N)\, \mathrm{d}\Omega + \int_{\Gamma_e} \hat{\psi}_i \hat{\mathbf{n}} \cdot (\mathbf{F}_N - \mathbf{F}_N^*)\, \mathrm{d}\Gamma, \tag{16}$$

where $\hat{\psi}_i = \sum_{k=1}^{M_N} M_{ik}^{-1} \psi_k$ with the mass matrix $M_{ik} = \int_{\Omega_e} \psi_i \psi_k \mathrm{d}\Omega$; for the sake of simplicity, we did not write the dependence on $\boldsymbol{x}$ of the basis functions although it should be understood that the basis functions depend on the spatial coordinates defined at both the interpolation and integration points. Note that $\hat{\psi}$ is constructed as a matrix-vector product between the inverse of the mass matrix M^{-1} and the column-vector of basis functions ψ. The inverse of the $M_N \times M_N$ matrix M is constructed via Gauss-Jordan and only needs to be done once (at the beginning of the simulation). Furthermore, if we maintain the same polynomial order N throughout all the elements Ω_e in the domain $\Omega = \bigcup_{e=1}^{N_e} \Omega_e$ and if we insist that the elements have straight edges, then the matrix M^{-1} needs to be calculated for only one canonical element (in the computational space) and then scaled by the Jacobian of the element Ω_e. This allows for a very simple and efficient construction of the key matrices required in an adaptive DG simulation.

The numerical flux function F^* is approximated by the Rusanov flux, given by:

$$\mathbf{F}_N^* = \frac{1}{2}\left[\mathbf{F}\left(\boldsymbol{q}_N^L\right) + \mathbf{F}\left(\boldsymbol{q}_N^R\right) - \lambda\,\hat{\mathbf{n}}\left(\boldsymbol{q}_N^R - \boldsymbol{q}_N^L\right)\right] \tag{17}$$

with the maximum wave speed $\lambda = ||\boldsymbol{u}||_2 + a$ where $||\boldsymbol{u}||_2 = \sqrt{u^2 + w^2}$ and a is the speed of sound. The superscripts L and R stand for the left and right limiting values at the boundary of the element. If the normal vector $\hat{\mathbf{n}}$ of element Ω_e is pointing to the right, $\boldsymbol{q}_N^L$ is the left limiting value of $\boldsymbol{q}_N$ and $\boldsymbol{q}_N^R$ is the right limiting value.

So far we have derived a discontinuous Galerkin discretization for our set of equations (10). At this point, the right hand side of eq. (16) is known and we can integrate the equation in time. This can be done either by an explicit or implicit method. For an explicit method we implement a third order Runge-Kutta method of Cockburn and Shu [27]. Because of the fast sound and gravity waves this explicit time-integration is restricted to a very short time-step. As explained before we are not interested in simulating these fast waves accurately; therefore, we also use an IMEX-method as presented in the next subsection.

3.2. IMEX-Method for Time Integration

The IMEX-method is implemented in a similar fashion to the approach of Restelli and Giraldo [28, 29]. The main difference is that we use potential temperature instead of total energy as the fourth variable.

The full nonlinear operator $\mathcal{N}(\boldsymbol{q})$ is given in our notation by

$$\mathcal{N}(\boldsymbol{q}) = -\boldsymbol{\nabla}\cdot\mathbf{F}(\boldsymbol{q}) + \boldsymbol{S}(\boldsymbol{q}). \tag{18}$$

For the IMEX-method we define an operator $\mathcal{L}$ by

$$\mathcal{L}\boldsymbol{q} = -\begin{pmatrix} \boldsymbol{\nabla}\cdot\boldsymbol{U} \\ \partial p'/\partial x \\ \partial p'/\partial z + g\,\rho' \\ \boldsymbol{\nabla}\cdot\left(\bar{\Theta}\,\boldsymbol{U}/\bar{\rho}\right) \end{pmatrix}, \tag{19}$$

where $p' = p - \bar{p}$ with $p = p(\Theta)$ and $\bar{p} = p(\bar{\Theta})$. The operator $\mathcal{L}$ and the term p' are linear with respect to $\left(\rho', \boldsymbol{U}^{\mathrm{T}}, \Theta'\right)$. As explained by Restelli [30] the operator $\mathcal{L}$ is responsible for the fast moving sound and gravity waves and, therefore, must be integrated implicitly. This splitting is done by writing

$$\frac{\partial\boldsymbol{q}}{\partial t} = \{\mathcal{N}(\boldsymbol{q}) - \mathcal{L}\,\boldsymbol{q}\} + \mathcal{L}\,\boldsymbol{q}. \tag{20}$$

For discretizing (20) in time, we use a backward difference formula of order 2, that leads to

$$\frac{1}{\gamma\,\Delta t}\sum_{m=-1}^{1}\alpha_m\boldsymbol{q}^{n-m} = \sum_{m=0}^{1}\beta_m\left[\mathcal{N}\left(\boldsymbol{q}^{n-m}\right) - \mathcal{L}\,\boldsymbol{q}^{n-m}\right] + \mathcal{L}\,\boldsymbol{q}^{n+1} \tag{21}$$

with $\alpha_{-1} = 1$, $\alpha_0 = 4/3$, $\alpha_1 = -1/3$, $\gamma = 2/3$, $\beta_0 = 2$, $\beta_1 = -1$ and Δt is the time step. We rewrite this equation collecting all terms with $\boldsymbol{q}^{n+1}$ and get

$$\left[\mathbf{I} - \gamma\,\Delta t\,\mathcal{L}\right]\boldsymbol{q}^{n+1} = \tilde{\boldsymbol{q}}^{\mathrm{ex}} - \gamma\,\Delta t\sum_{m=0}^{1}\beta_m\mathcal{L}\,\boldsymbol{q}^{n-m}, \tag{22}$$

where

$$\tilde{\boldsymbol{q}}^{\mathrm{ex}} = -\sum_{m=0}^{1}\alpha_m\boldsymbol{q}^{n-m} + \gamma\,\Delta t\sum_{m=0}^{1}\beta_m\mathcal{N}\left(\boldsymbol{q}^{n-m}\right) \tag{23}$$

is an explicit predictor that has to be calculated first. The identity matrix $\mathbf{I}$ on the left hand side of eq. (22) comes from the coefficient α_{-1}. Solving the linear system of equations (22) (e.g., with GMRES) gives the implicit corrector. For details on this solution strategy the reader is referred to [29] and [31].

The process of solving the linear system of equations can be improved by using preconditioners. Currently we just use the simple Jacobi preconditioner. For this preconditioner we found, that the simulation using the IMEX-method is still most efficient when using a time step which is about twice as large as for the explicit simulation. The time step could be much larger but with this simple preconditioner a larger time step takes more CPU time. We are working on developing better preconditioners [32]. We expect to be able to reduce the CPU time based on future research on specialized preconditioners.

3.3. Mesh Refinement with Space Filling Curve Approach

As explained in section 2 we expect steep gradients at the boundary of the cumulus clouds for which our numerical model is designed. For being able to increase the spatial resolution in these regions we use h-adaptive mesh refinement. This means that spatial resolution is adapted by dividing elements into smaller elements (refinement) or collecting elements into larger elements (coarsening). In our code this is managed with the function library AMATOS [11]. The main advantage of this function library is that it handles the entire h-adaptive mesh refinement. Furthermore it leads to an order of unknowns that preserves data adjacency and linearizes data access in computer memory by using a so-called space filling curve approach. For further information we refer to the publication of Behrens et al. [11].

The only modification that was necessary for our work was the calculation of the new values at the grid points when elements are refined or coarsened. This is quite straight forward. We simply evaluate the old polynomials at the positions of the new degrees of freedom.

The refinement criterion used for the results presented in this paper is given by:

$$|\theta'(\boldsymbol{x},t)| \geq \sigma \max_{\boldsymbol{x}} \left(|\theta'(\boldsymbol{x},t)|\right), \tag{24}$$

with the deviation of the potential temperature from the background state denoted by $\theta' = \theta - \bar{\Theta}/\bar{\rho}$ and a user-specified and problem dependent constant σ. For the density current of Straka et al. [33] we use $\sigma = 0.05$. In all the other results shown in this paper we used $\sigma = 0.1$. Wherever this condition (24) is fulfilled the mesh is refined until it reaches a specified finest resolution. In the rest of the domain the grid is coarsened until it reaches a specified coarsest resolution without modifying the resolution in the refinement region. The transition between fine and coarse meshes is given by the conformity of the grid. After each time-step we calculate the number of elements that have to be changed for grid refinement. If more than 1% of all elements have to be changed, the grid is adapted. Otherwise the grid remains unchanged.

To avoid small scale structures moving into a region with a coarse mesh we add a few rows of fine elements to the refinement region. We do not know a priori which size of the refinement region is best. For the verification in section 4 we choose the size of this buffer zone as we expect it to be the best. For getting a more objective instruction on how to choose the size of the refinement region we develop a new method for testing refinement criteria in section 5. In that section we will also compare the results for different sizes of this buffer zone.

The time-step is determined by the smallest grid spacing. So far we do not have any sub-cycling implemented, because we expect that for the simulation of cumulus clouds the majority of elements will be very small. For this reason we do not expect much benefit by using sub-cycling in these applications.

The refinement criterion (24) should not be used for every application. For the simple test cases shown in the rest of this paper we will see that it works very well. However, for more realistic simulations, we intend to use better refinement criteria, based on physical parameters, gradients in fluid components and mathematical error indicators.

3.4. Artificial viscosity

For simulating the density current test case of Straka et al. [33] we implemented the artificial viscosity that is used in [33]. A diffusion term $\boldsymbol{\nabla} \cdot (\mu\,\rho\,\boldsymbol{\nabla}\boldsymbol{u})$ is added to the right hand side of eq. (2) and the term

$\nabla \cdot (\mu \, \rho \, \nabla \theta')$ is added to the right hand side of eq. (3), with μ the viscosity parameter. We use the deviation of potential temperature θ' in the viscosity term because we do not want viscosity to modify the hydrostatic basic state. An example for the application of these viscosity terms to stratified flow is given by Gabersek et al. [34].

In addition to the use of artificial viscosity for simulating the test case of Straka et al. we use the diffusion terms with a non-constant viscosity parameter μ_{lim} as a limiter for reducing the Gibbs phenomenon. The viscosity parameter μ_{lim} is still constant within each element of our grid but we allow μ_{lim} to change between different elements. The total viscosity parameter for each element e is given by

$$\mu_e = \max\left(\mu_{\text{tc}}, \mu_{\text{lim},e}\right), \tag{25}$$

where μ_{tc} is the viscosity parameter given by the test case (e.g. $\mu_{\text{tc}} = 75\text{m}^2/\text{s}$ for the density current of Straka et al. [33]). For the parameter $\mu_{\text{lim},e}$ we introduced the following formula:

$$\mu_{\text{lim},e} = \mu_{\text{ref}} \frac{\Delta\theta'_e}{\alpha\, \Delta\theta'_0} \frac{\Delta x_{\text{eff}}}{\Delta x_{\text{ref}}} \frac{v_{\text{max}}}{v_{\text{ref}}}, \tag{26}$$

where $\Delta\theta'_e$ is the difference between the maximum and minimum of θ' in element e, $\Delta\theta'_0$ is the difference between the maximum and minimum of θ' at time $t = 0$s taken over the whole domain, Δx_{eff} is the effective spatial resolution as defined in section 3.1, v_{max} is the approximate maximum wind speed throughout the whole simulation, and μ_{ref}, α, κ, Δx_{ref} and v_{ref} are fixed parameters. As $\Delta\theta'_e / \Delta x_{\text{eff}}$ is a measure for the gradient of potential temperature, the proportionality of $\mu_{\text{lim},e}$ to a gradient and to Δx^2_{eff} in equation (26) is similar to the approach for artificial viscosity introduced by VonNeumann and Richtmyer [35].In future research we will consider more recent approaches for using artificial viscosity like those by [36–38].

We tested different choices of parameters and found that $\mu_{\text{ref}} = 0.1\,\text{m}^2/\text{s}$, $\alpha = 0.4$, $\Delta x_{\text{ref}} = 3.12\,\text{m}$ and $v_{\text{ref}} = 3\,\text{m/s}$ give us the best compromise between damping of the Gibbs phenomenon and preserving the amplitude of the flow for all test cases considered in this paper. The maximum wind speed v_{max} was estimated by making high-resolution simulations of the test cases. For the density current test case from Straka et al. [33] we found $v_{\text{max}} \approx 40\,\text{m/s}$. The other test cases presented in this paper show a maximum wind speed of about $v_{\text{max}} \approx 3\,\text{m/s}$. In order to avoid steep gradients moving into elements with a low viscosity coefficient we compute μ_{lim} according to (26) and take the maximum value of all neighboring elements. For the high-resolution run for the test case of Giraldo and Restelli in figure 7d we found that the maximum should be taken over two rows of neighboring elements.

This approach does not remove the Gibbs phenomenon completely. For dry inviscid atmospheric convection potential temperature is conserved. This allows us to filter most of the Gibbs phenomenon by using the following cutoff-filter $\mathcal{F}$ at any of our degrees of freedom i:

$$\mathcal{F}(\theta_i) = \begin{cases} \theta_{\text{max},0}, & \text{if } \theta_i > \theta_{\text{max},0}, \\ \theta_i, & \text{if } \theta_{\text{max},0} \geq \theta_i \geq \theta_{\text{min},0}, \\ \theta_{\text{min},0}, & \text{if } \theta_i < \theta_{\text{min},0}, \end{cases} \tag{27}$$

where $\theta_{\text{min},0}$ and $\theta_{\text{max},0}$ are the global minimum and maximum of potential temperature θ at time $t = 0\,$s. For the test cases shown in this paper we always use this simple filter and we will see that it works very well in combination with artificial viscosity. Nevertheless we are working on implementing and testing other limiting techniques.

The second order derivatives produced by the diffusion terms are discretized by using a local discontinuous Galerkin method [39]. That means that the order of the derivatives is reduced by introducing the following new variables

$$\boldsymbol{\alpha} = \nabla u, \tag{28}$$
$$\boldsymbol{\beta} = \nabla w, \tag{29}$$
$$\boldsymbol{\gamma} = \nabla \theta'. \tag{30}$$

Similar to eq. (16) we get

$$\boldsymbol{\alpha}_i = -\int_{\Omega_e} \hat{\psi}_i \, u_j \boldsymbol{\nabla} \psi_j \mathrm{d}\Omega + \int_{\Gamma_e} \hat{\psi}_i \psi_j \hat{\mathbf{n}} \left(u_j - u_j^* \right) \mathrm{d}\Gamma, \tag{31}$$

$$\boldsymbol{\beta}_i = -\int_{\Omega_e} \hat{\psi}_i \, w_j \boldsymbol{\nabla} \psi_j \mathrm{d}\Omega + \int_{\Gamma_e} \hat{\psi}_i \psi_j \hat{\mathbf{n}} \left(w_j - w_j^* \right) \mathrm{d}\Gamma, \tag{32}$$

$$\boldsymbol{\gamma}_i = -\int_{\Omega_e} \hat{\psi}_i \, \theta_j' \boldsymbol{\nabla} \psi_j \mathrm{d}\Omega + \int_{\Gamma_e} \hat{\psi}_i \psi_j \hat{\mathbf{n}} \left(\theta_j' - {\theta_j'}^* \right) \mathrm{d}\Gamma. \tag{33}$$

As these viscosity terms do not describe a flow in a certain direction (as in the case of the advection terms) we use the following numerical flux for the viscosity terms in $\mathbf{F}_N^*$:

$$Q^* = \frac{1}{2} \left(Q^R + Q^L \right) \tag{34}$$

where Q represents either u, w or θ'. Note that this is not the only possibility for discretizing the second order operators; for other choices see Shahbazi et al. [40].

4. Verification

For the verification of our numerical model we considered three test cases that are relevant for atmospheric convection. These test cases are a small cold air bubble on top of a large warm air bubble from Robert [41], a density current from Straka et al. [33], and a smooth warm air bubble from Giraldo and Restelli [22]. In all of these cases there is no exact solution, but we can compare our results with those from the literature.

The results of adaptive simulations of the different test cases are shown in figures 2–7. Some more details of the simulations are given in tables 1–3. The tables show also the details of uniform simulations using the finest spatial resolution of the corresponding adaptive computation. The values suggest that adaptive simulations are much faster compared to the uniform simulations. However at this point it is not completely clear how large the refinement region should be. For this purpose we compare adaptive and uniform simulations in section 5.

4.1. Small Cold Air Bubble on Top of Large Warm Air Bubble

The first test case is a small cold air bubble on top of a large warm air bubble in a domain of 1km×1km. This test case was introduced by Robert [41] in 1993. The background state has a constant potential temperature of $\bar{\theta} = 303.15\,\mathrm{K}$. Both bubbles have a Gaussian profile in θ'. The warm air bubble has an amplitude of 0.5 K, the amplitude of the cold air bubble is $-0.15\,\mathrm{K}$. The initial conditions are chosen identically to those of Robert [41]. However, we use a slightly different resolution. This is necessary because in our case the domain has to be divided into a hierarchy of triangles for adaptive grid refinement. Therefore the resolution can only be changed by a factor of $\sqrt{2}$. In this test case we use a resolution of $\Delta x_{\mathrm{eff}} = 4.4\,\mathrm{m}$ (table 1) which is slightly smaller than 5m of Robert [41].

Figure 2 shows our result for this test case. The mesh is continuously adapted to the position of the temperature anomaly. Correspondingly the fine mesh follows the bubbles very nicely. As mentioned before we use polynomials of degree three for all results shown in this paper.

By comparing our result with the corresponding figure of Robert [41] one can see that the results agree very well. After 600s the position and shape of the warm air is still almost identical to the corresponding plot of Robert [41]. Even the smaller vortices on the right hand side of the domain are very similar to the result of Robert.

4.2. Density Current

A second test case is a density current initialized by a cold air bubble with a cosine profile in temperature T and an amplitude of -15 K, which leads to an amplitude of -16.624 K in θ' (figure 3). This test case was introduced by Straka et al. [33]. The viscosity of $\mu_{\mathrm{tc}} = 75\mathrm{m}^2/\mathrm{s}$ is identical to the setup of Straka et al. [33].

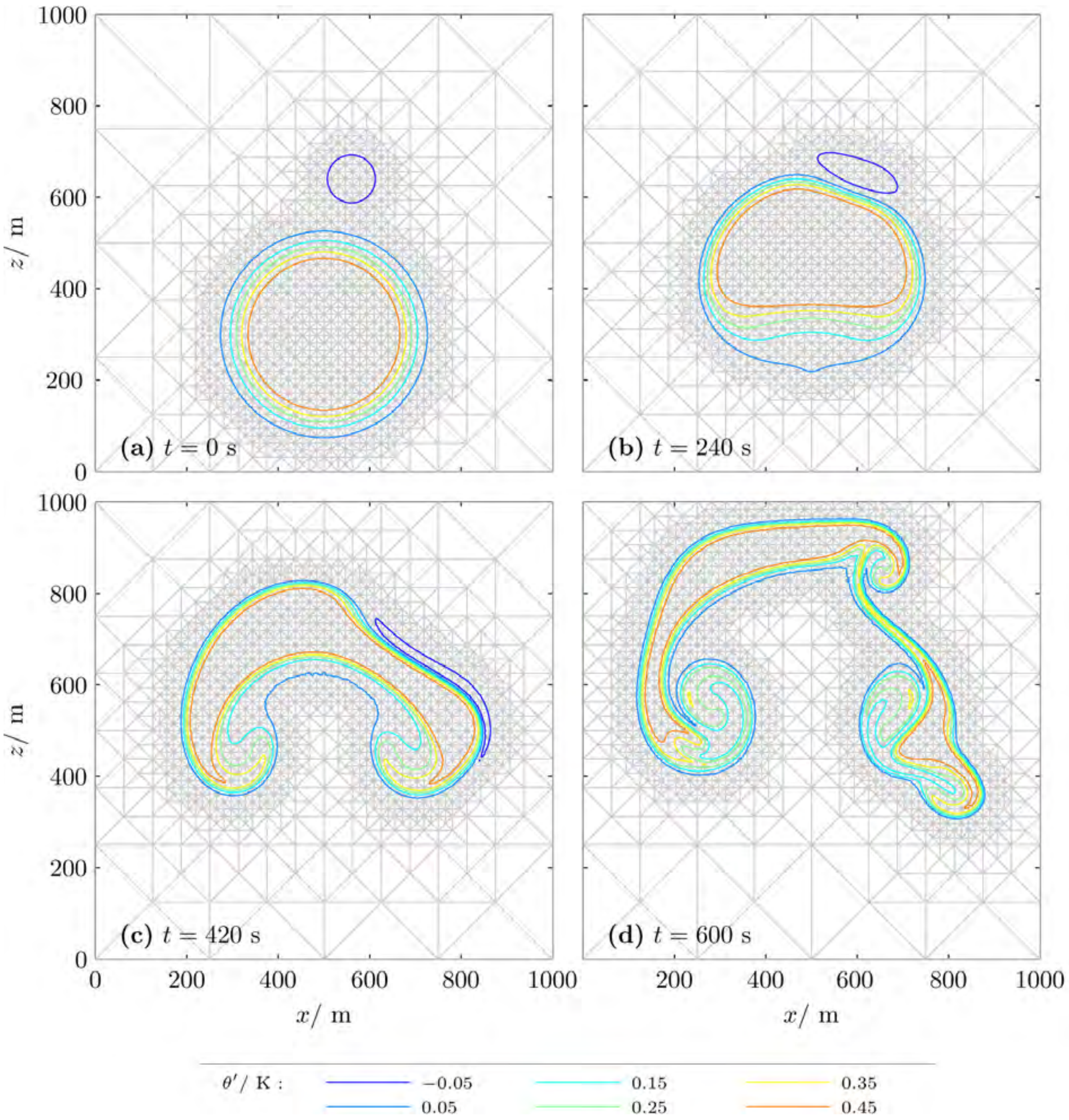

Figure 2: Small cold air bubble on top of a large warm air bubble as introduced by Robert [41]. The contour lines show the deviation of the potential temperature from the background state and the gray lines show the adaptively refined triangular mesh used in our simulation. The contour values are from -0.05 K to 0.45 K with an interval of 0.1 K. For the time-integration we use the IMEX-method for all simulations shown in this paper. We also tested explicit time-integration which produces the same results.

	adaptive	uniform
Δt	0.020 s	0.020 s
Δx_{eff}	4.42 m	4.42 m
L	15.62 m	15.62 m
#elements	2833.56	8192
total time	10857.04 s	32804.87 s
grid time	721.71 s	0.16 s
max(θ')	0.50 K	0.50 K
min(θ')	-0.05 K	-0.04 K
figure	2	—

Table 1: Details for our simulations of a bubble test case by Robert (1993) [41]. The variables shown in this table are: the time step Δt, the finest effective resolution Δx_{eff}, the length of the shortest element edge L, the average number of elements for the whole simulation, the total CPU time, the CPU time spent for initializing and adapting the grid ("grid time"), the maximum and minimum of θ' at the end of the simulation ($t = 600$s) and the figure where the simulation is shown. The number of elements is an average value from time $t = 0$ s to end time $t = 600$ s. Therefore it is an integer value for the uniform simulation but not for the adaptive simulation. The CPU times give the time until the end of the simulation at $t = 600$ s is reached. All simulations use the IMEX-method described in section 3.2 for time-integration and are computed on the same single Linux CPU.

As described for the previous test case we are not able to use exactly the same resolution as in the literature. For the high-resolution run shown in figure 3 we use $\Delta x_{\text{eff}} = 28.26$m which is slightly larger than the reference resolution of Straka et al. with 25m. Again we see no differences between our result and the result in the literature. The position of the density current and the shape of the Kelvin-Helmholtz rotors agrees very well with the result of Straka et al. [33] throughout the whole simulation. As given in table 2 the horizontal location of the front at time $t = 900$ s is $x_{\text{front}} = 15452.40$ m while the reference simulation of Straka et al. gives a position of $x_{\text{front}}^{\text{Straka}} = 15537.44$ m.

Figure 4 shows the corresponding result for a much coarser resolution of 226.08 m. The position of the front $x_{\text{front}} = 15116.89$ m and the amplitude of the density current of -9.24 K agree fairly well with the highly-resolved simulations. Figure 5 shows the result for an effective resolution of 452.16 m. Compared to the numerical models presented in [33] we think these results reach at least the average quality of the different results presented in that publication.

When decreasing our limiter parameter μ_{ref} the front position and amplitude of the density current get even closer to the high-resolution result in figure 3. If we set $\mu_{\text{ref}} = 0.05\,\text{m}^2/\text{s}$ instead of $\mu_{\text{ref}} = 0.1\,\text{m}^2/\text{s}$ we get the result shown in figure 6. In this result the position of the front $x_{\text{front}} = 15414.7$ m and the amplitude of the density current -10.50 K are very close to the high-resolution result. The noise in figure 6 is slightly more pronounced than in figure 4, but it is still very reasonable. This demonstrates that small changes in μ_{ref} could be used for adapting the simulation to the degree of grid noise that can be allowed in a certain application. Nevertheless our results demonstrate that even a fixed value of μ_{ref} produces satisfactory results in all test cases.

4.3. Smooth Warm Air Bubble

As a third test case we computed the rising thermal bubble introduced by Giraldo and Restelli [22] (test case 2). It is a single warm air bubble with a cosine profile in θ'. As in the test case in section 4.1 the domain has an extent of 1 km in each direction and the bubble has an amplitude of 0.5 K. We use the same parameters as in the publication of Giraldo and Restelli [22] except for the slightly different resolutions as given in table 3. Furthermore we replaced the Boyd-Vandeven type filter of Giraldo and Restelli [22] with our artificial viscosity based limiter.

As in the previous test cases there are no obvious differences between our results (figure 7) and those from the literature. The position and shape of the bubble seems to be identical to the result of Giraldo and Restelli [22]. This gives us confidence that our code is free of fundamental errors.

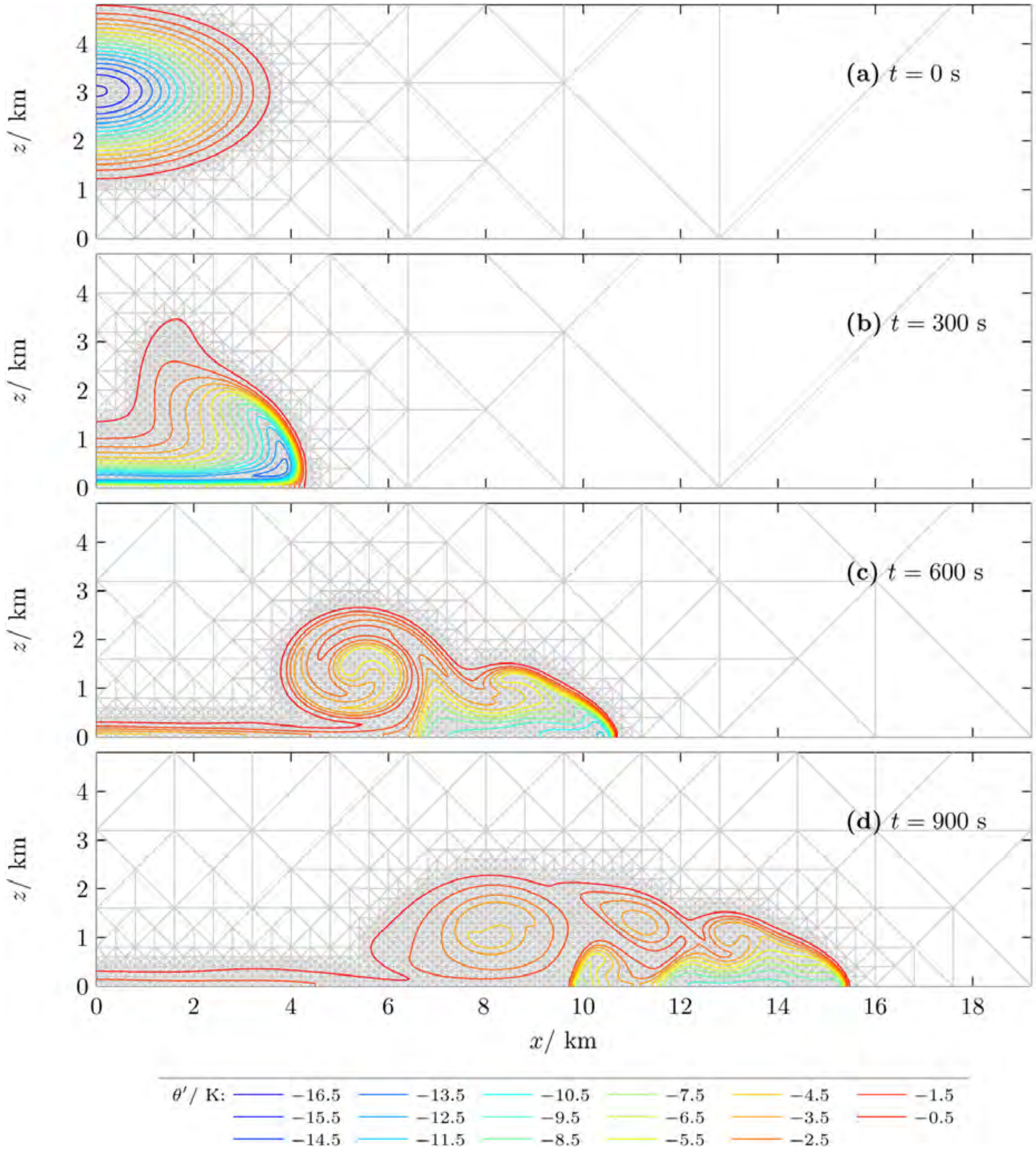

Figure 3: Density current as introduced by Straka et al. [33] for an effective resolution of $\Delta x_{\text{eff}} = 28.26\,\text{m}$. As in figure 2 the contour lines show the deviation θ' of the potential temperature θ from the background state $\bar{\theta}$ and gray lines show the adaptively refined triangular mesh. For making it easier to compare our results with those in [33] we use the same contour values, which are from -16.5 K to -0.5 K with an interval of 1 K.

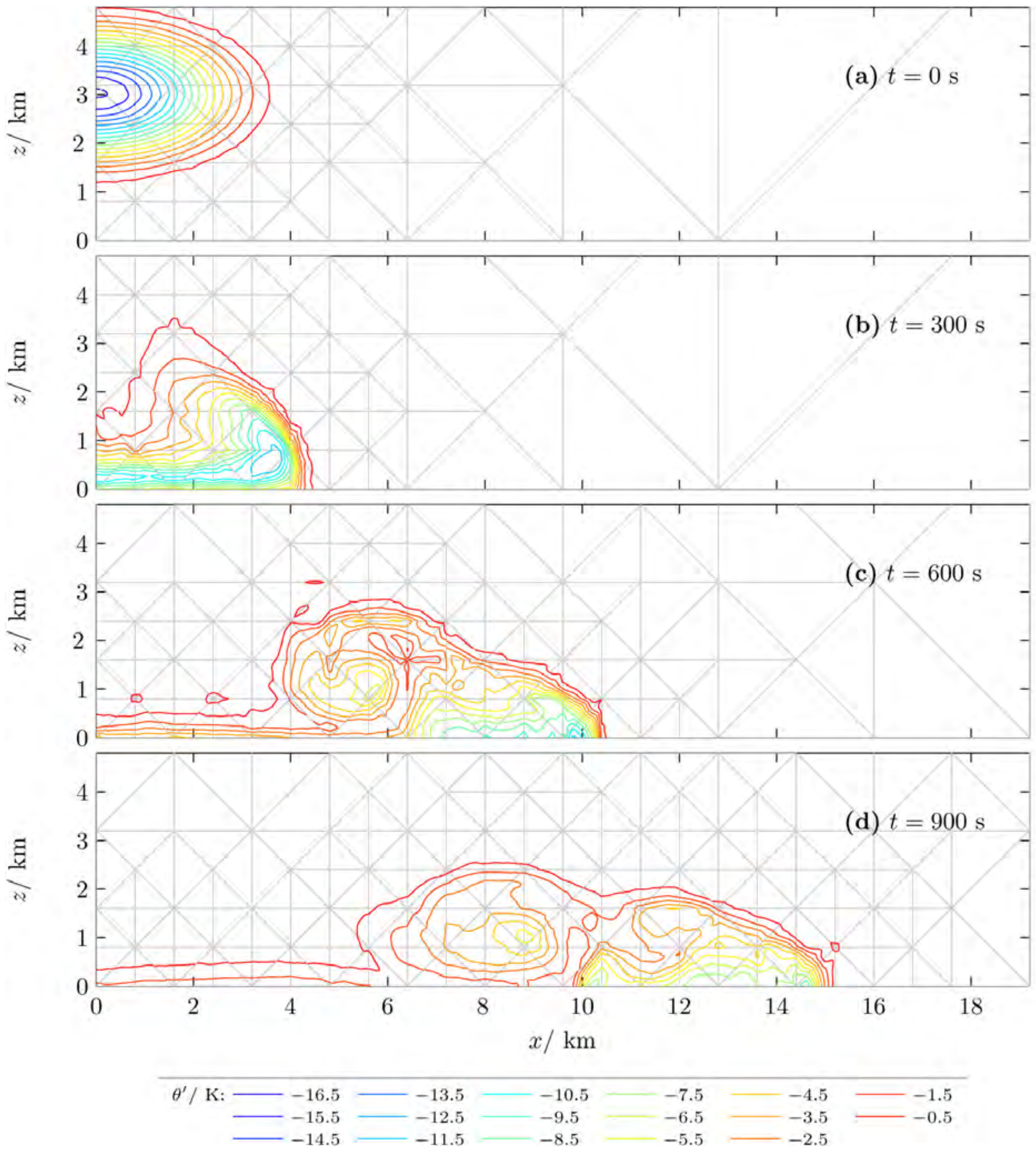

Figure 4: Density current as in figure 3 for an effective resolution of $\Delta x_{\text{eff}} = 226.08\,\text{m}$.

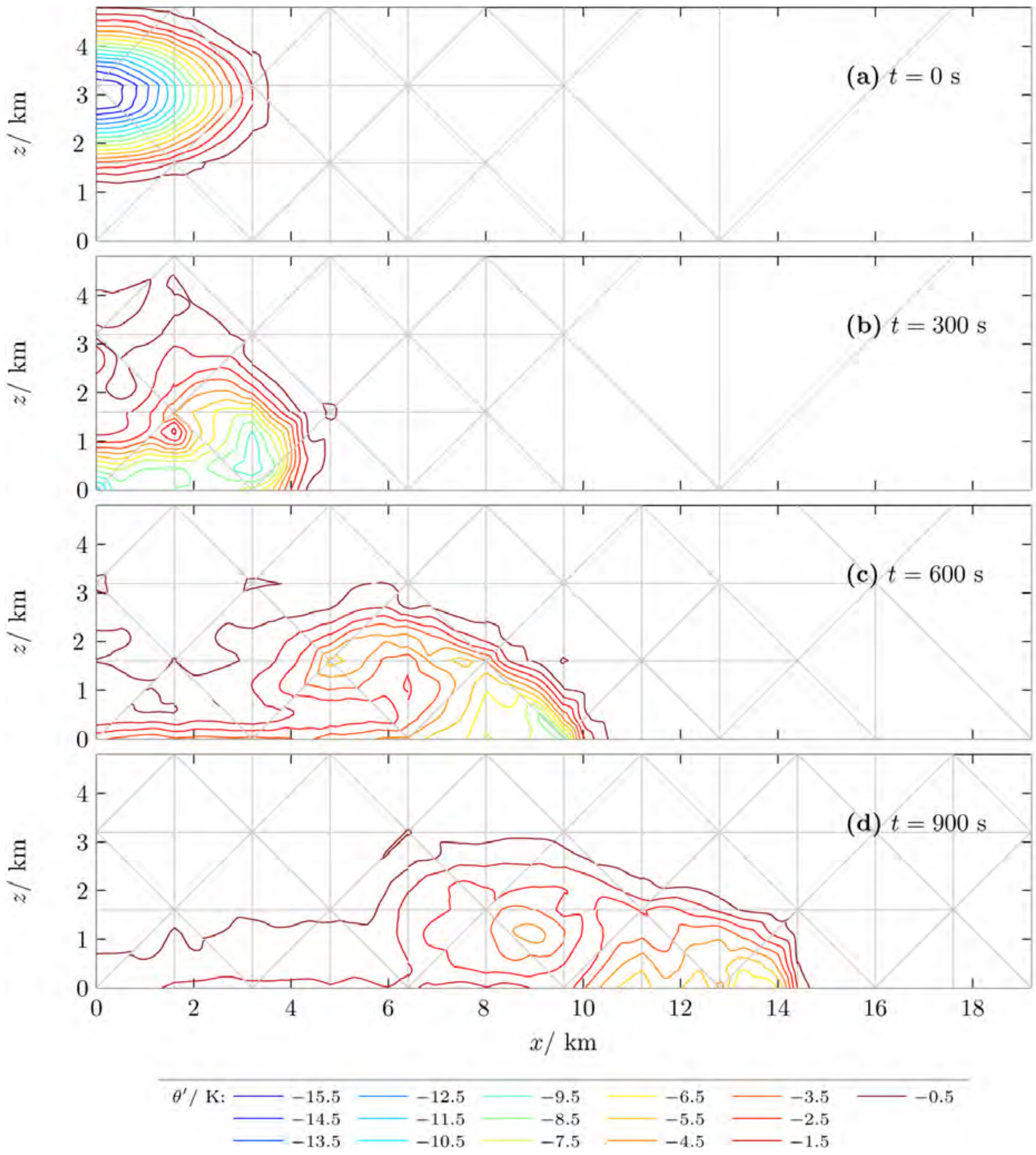

Figure 5: Density current as in figure 3 for an effective resolution of $\Delta x_{\text{eff}} = 452.16\,\text{m}$.

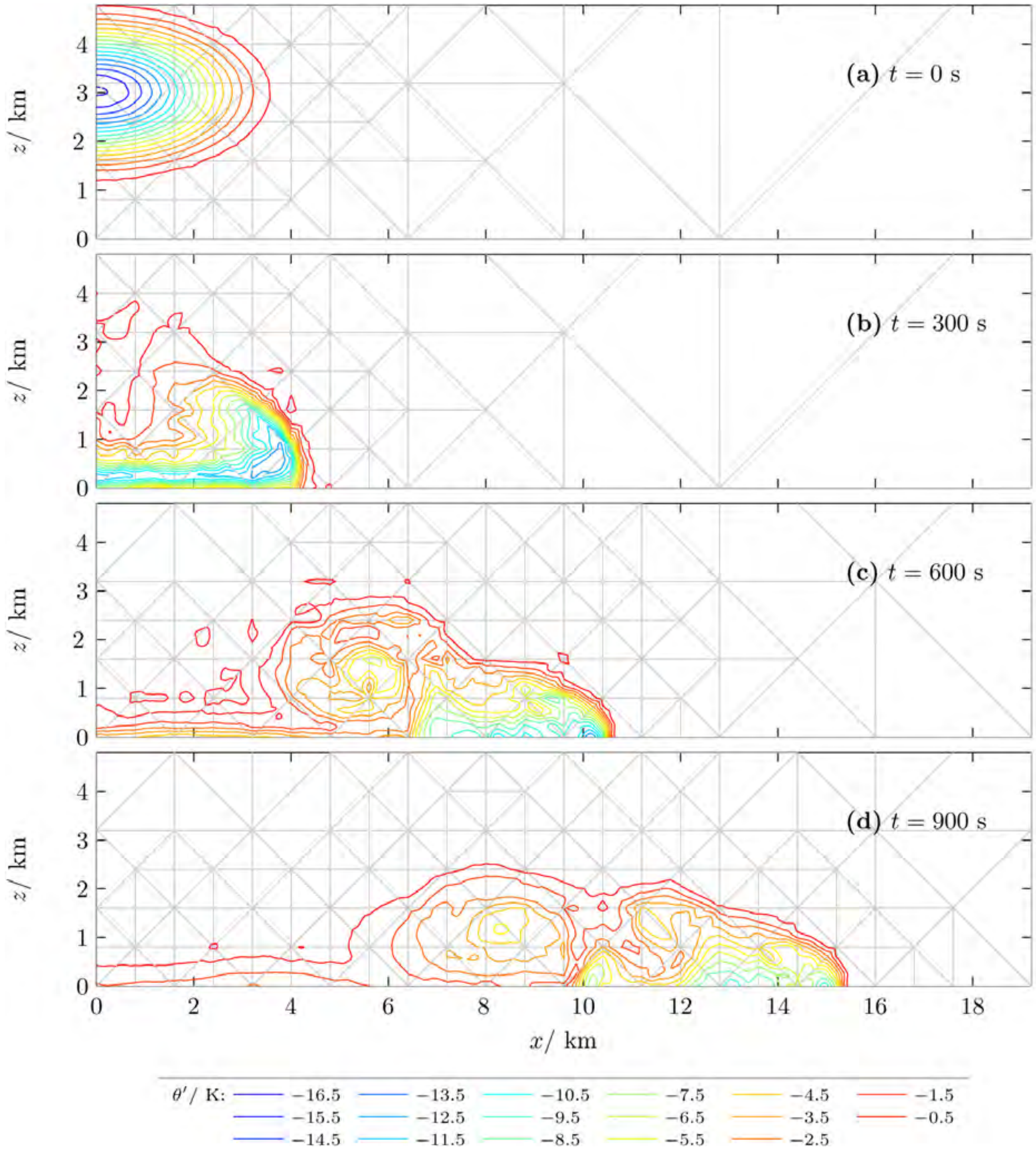

Figure 6: Density current as in figure 4 for a modified limiter parameter of $\mu_{\text{ref}} = 0.05\,\text{m}^2/\text{s}$. The spatial resolution is $\Delta x_{\text{eff}} = 226.08\,\text{m}$

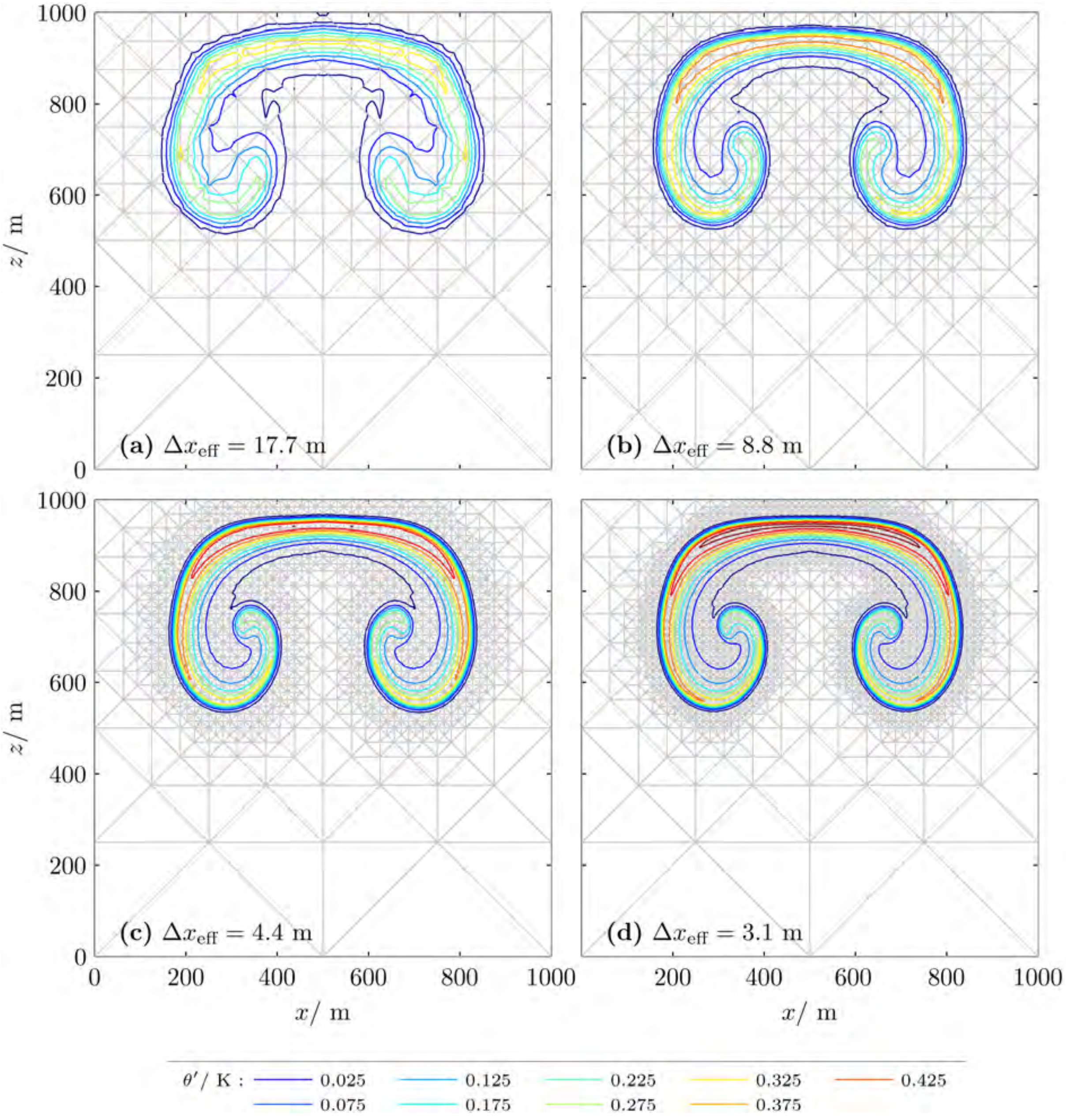

Figure 7: Rising thermal bubble introduced by Giraldo and Restelli [22] at time $t = 700$s. As in figure 2 the contour lines show the deviation θ' of the potential temperature θ from the background state $\bar{\theta}$ and gray lines show the adaptively refined triangular mesh. Contour values are from 0.025 K to 0.425 K with an interval of 0.05 K.

	Straka 1	Straka 2	Straka 3	Straka 4	Straka 5	Straka 6
adaptive	yes	no	yes	no	yes	no
Δt	0.100 s	0.100 s	1.000 s	1.000 s	2.000 s	2.000 s
Δx_{eff}	28.26 m	28.26 m	226.08 m	226.08 m	452.16 m	452.16 m
L	100.00 m	100.00 m	800.00 m	800.00 m	1600.00 m	1600.00 m
#elements	3197.80	32768	151.56	512	64.75	128
total time	3615.99 s	26905.62 s	23.11 s	61.79 s	4.90 s	8.16 s
grid time	327.37 s	0.44 s	1.02 s	0.01 s	0.20 s	0.00 s
$\max(\theta')$	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K
$\min(\theta')$	−9.80 K	−9.80 K	−9.24 K	−9.17 K	−7.22 K	−7.22 K
front position:	15452.40 m	15442.63 m	15116.89 m	15112.64 m	14525.40 m	14521.57 m
figure	3	—	4	—	5	—

Table 2: Details for our simulations of the density current test case by Straka et al. (1993) [33]. The different columns represent different setups of spatial resolution and adaptivity. The variables shown in this table are described in the caption of table 1. The end time of the simulation is $t = 900$s (instead of $t = 600$s in table 1). Additionally the horizontal position of the density current front at time $t = 900$s (given by the -1K contour) is denoted.

	Giraldo 1	Giraldo 2	Giraldo 3	Giraldo 4	Giraldo 5	Giraldo 6	Giraldo 7	Giraldo 8
adaptive	yes	no	yes	no	yes	no	yes	no
Δt	0.050 s	0.050 s	0.030 s	0.030 s	0.020 s	0.020 s	0.010 s	0.010 s
Δx_{eff}	17.66 m	17.66 m	8.83 m	8.83 m	4.42 m	4.42 m	3.12 m	3.12 m
L	62.50 m	62.50 m	31.25 m	31.25 m	15.62 m	15.62 m	11.05 m	11.05 m
#elements	269.17	512	843.41	2048	2157.03	8192	3637.28	16384
total time	300.69 s	579.18 s	1804.15 s	4209.46 s	9134.92 s	33010.70 s	25171.27 s	102955.38 s
grid time	21.19 s	0.02 s	150.53 s	0.06 s	618.30 s	0.13 s	2119.41 s	0.36 s
$\max(\theta')$	0.32 K	0.32 K	0.36 K	0.36 K	0.42 K	0.42 K	0.44 K	0.44 K
$\min(\theta')$	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K	0.00 K
figure	7a	—	7b	—	7c	—	7d	—

Table 3: Details for our simulations of the warm air bubble test case by Giraldo and Restelli (2008) [22]. The different columns represent different setups of spatial resolution and adaptivity. The variables shown in this table are described in the caption of table 1. The end time of the simulation is $t = 700$s (instead of $t = 600$s in table 1).

5. Comparison Between Adaptive and Uniform Simulations

So far we used adaptive mesh refinement without exactly knowing how large the refinement region should be for producing a similar accuracy as achieved in uniform simulations. This section investigates the following three questions for the warm air bubble test case from section 4.3:

1. Is it possible to find a refinement region for which the adaptive simulation produces exactly the same result like a uniform simulation which uses the finest spatial resolution of the adaptive simulation?
2. Which size of refinement region is needed for avoiding significant additional errors by using adaptive mesh refinement?
3. For a given CPU-time: does the adaptive simulation produce a more accurate result than a uniform simulation?

The first two questions assume that the uniform simulation uses the finest resolution of the adaptive simulation for being able to simulate the same small scale features of the flow. However for resolving small scale turbulence in a cloud a spatial resolution of about 0.1 mm has to be used [20]. In most atmospheric applications such a fine resolution is far beyond todays computer capacity. For this reason the third question is very important for atmospheric applications.

These three questions are considered in this section for the warm air bubble test case of Giraldo and Restelli [22]. For avoiding the creation of more and more small scale turbulence with increasing resolution we use a constant amount of artificial viscosity. If not otherwise noted we use $\mu_{\mathrm{tc}} = 0.1\,\mathrm{m}^2/\mathrm{s}$ for the viscosity parameter. In this section the viscosity is considered to be part of the test case.

In the following three subsections we consider these questions. The titles of the subsections repeat the questions in a shortened form.

5.1. *Can Adaptive and Uniform Results be Identical?*

In this subsection we consider the question whether it is possible to achieve exactly the same result with an adaptive simulation as with a uniform simulation which uses the finest resolution of the adaptive simulation. For a first overview figure 8 and 9 show the absolute value of the difference between adaptive and uniform warm air bubble simulations for different sizes of the refinement region. All simulations in figure 8 and 9 use the refinement criterion (24). However the number of additional fine elements surrounding the region where (24) is fulfilled differs.

The smallest refinement region (figure 8a) does not cover the warm air bubble at time $t = 0\,\mathrm{s}$ completely. For this reason it is not surprising that this simulation shows quite significant deviations from the corresponding uniform simulation. However even the fairly large refinement regions (figure 9c and d) show deviations at later times.

The question remains whether the difference between adaptive and uniform simulations is decreasing continuously with increasing size of the refinement region. For answering this question figure 10 shows the L^2-norm of the difference between adaptive and uniform simulations with a finest spatial resolution of $\Delta x_{\mathrm{eff}} = 4.4\,\mathrm{m}$ as a function of the average number of elements. At time $t = 200\,\mathrm{s}$ the L^2-norm is significantly larger when the simulation uses less than about 1500 elements. This behavior is reasonable because about 1500 elements are needed for covering the warm air bubble at time $t = 0\,\mathrm{s}$ completely. For more than 1500 elements the L^2-norm is approximately constant at time $t = 200\,\mathrm{s}$ except for some fluctuations when more than 3000 elements are used. We did not investigate these fluctuations in detail. It might be that they are produced by coarse elements in the edges of the domain.

As already mentioned figure 10 shows at time $t = 200\,\mathrm{s}$ a fairly clear distinction between simulations that are affected by using adaptive mesh refinement (with less than about 1500 elements) and others that are not significantly affected. However at time $t = 400\,\mathrm{s}$ it becomes very difficult to make such a distinction and at time $t = 600\,\mathrm{s}$ almost every adaptive simulation seems to be affected by using adaptive mesh refinement. In some way this result might be reasonable: the rising warm air bubble produces a circulation in the whole domain which cannot be represented on the coarse mesh of the adaptive simulations in the same way as in the uniform simulations. Nevertheless we did not expect that the L^2-norm is decreasing in such a continuous way. We expected that the L^2-norm of the difference between adaptive and uniform simulations decreases more quickly after reaching a certain size of the refinement region. At time $t = 800\,\mathrm{s}$ and $t = 1000\,\mathrm{s}$ there are some smaller jumps in the curves in figure 10. However these jumps vanish fairly soon again and they cannot be seen in the same way when using a finest resolution of $\Delta x_{\mathrm{eff}} = 3.1\,\mathrm{m}$ (figure 11).

Until $t = 600\,\mathrm{s}$ the behavior of the L^2-norm of the difference between adaptive and uniform simulations with a finest resolution of $\Delta x_{\mathrm{eff}} = 3.1\,\mathrm{m}$ (figure 11) is very similar to the results at $\Delta x_{\mathrm{eff}} = 4.4\,\mathrm{m}$ (figure 10). For $t \geq 800\,\mathrm{s}$ the difference between adaptive and uniform simulations is significantly larger when using a large refinement region than in the case of a finest resolution of $\Delta x_{\mathrm{eff}} = 4.4\,\mathrm{m}$ (figure 10). This might be explained by instabilities at the boundary of the warm air bubble which are better resolved in the higher resolution simulations and lead to a faster growth of the difference between adaptive and uniform simulations. We discuss the instabilities at the boundary of warm air bubbles in the next subsection more in detail.

The figures shown in this subsection demonstrate that adaptive simulations of our warm air bubble test case are not able to produce exactly the same result as uniform simulations which use the finest resolution of the adaptive simulation. Furthermore it is not clear how large the L^2-norm of the difference between the adaptive and uniform simulation should be for being negligible. Therefore we introduce a different approach in the next subsection.

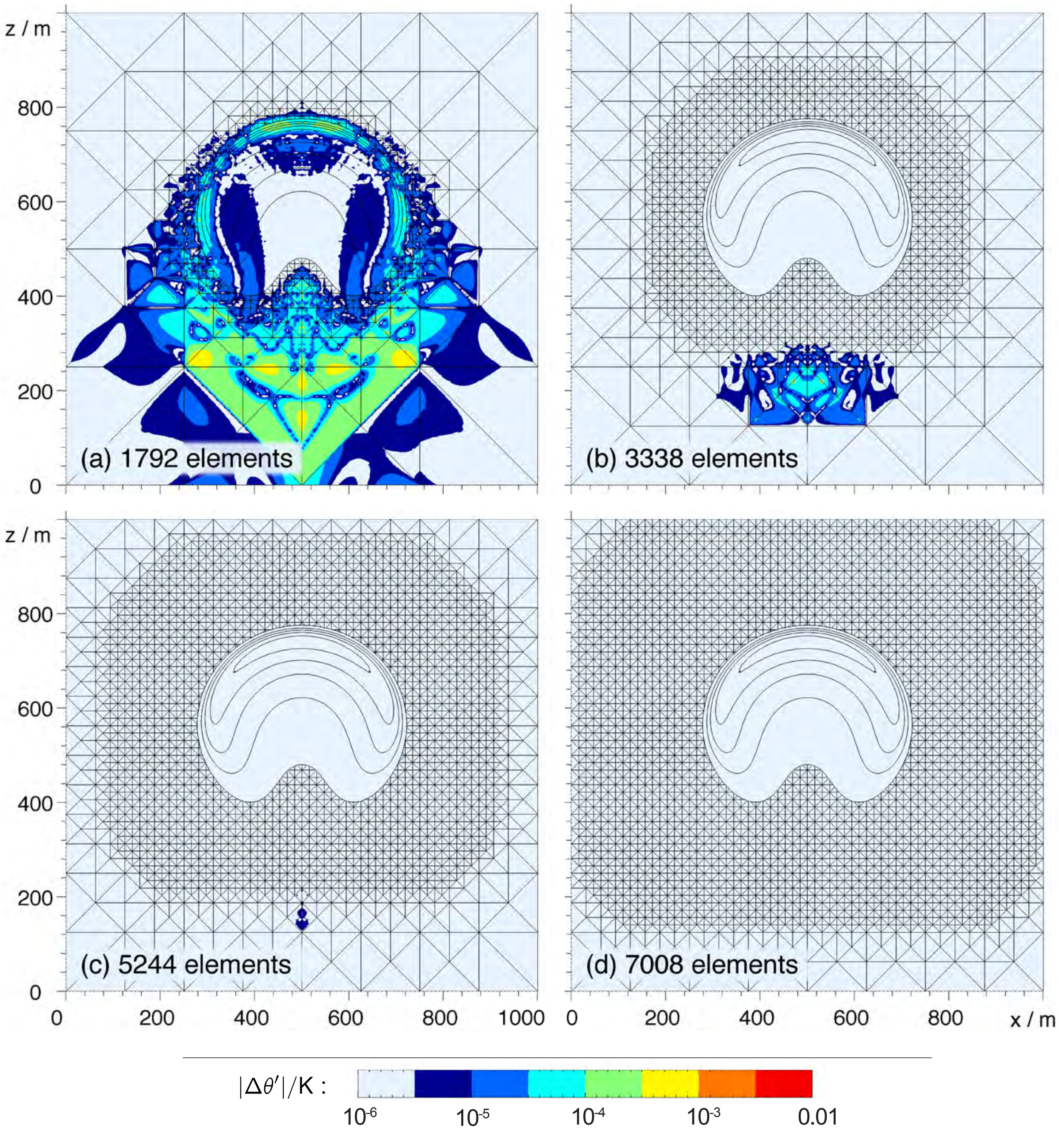

Figure 8: Comparison between adaptive and uniform simulations of the warm air bubble from section 4.3 with a constant artificial viscosity of $\mu_{\mathrm{tc}} = 0.1\,\mathrm{m}^2/\mathrm{s}$ at time $t = 400\,\mathrm{s}$ for four different sizes of the refinement region. The labels of the different subfigures show the number of elements used at time $t = 400\,\mathrm{s}$. Color shows the absolute value of the difference in θ' between an adaptive simulation with a finest spatial resolution of $\Delta x_{\mathrm{eff}} = 4.4\,\mathrm{m}$ and a uniform simulation with $\Delta x_{\mathrm{eff}} = 4.4\,\mathrm{m}$. The colormap is logarithmic. The smallest value of the colormap was chosen to be $|\Delta\theta'| = 10^{-6}\,\mathrm{K}$ for making it easier to see the regions with significant values of $|\Delta\theta'|$. The largest value of the colormap is $|\Delta\theta'| = 0.01\,\mathrm{K}$ because this is the maximum value of $|\Delta\theta'|$ at time $t = 800\,\mathrm{s}$ (see figure 9). Straight black lines show the locally refined mesh of the adaptive computation. The mesh is not shown where $\theta' > 0.05\,\mathrm{K}$ for making it easier to see the position of the warm air bubble. Black contour lines show the deviation of potential temperature from the background state θ' in the adaptive simulation from $0.05\,\mathrm{K}$ to $0.45\,\mathrm{K}$ with an interval of $0.1\,\mathrm{K}$.

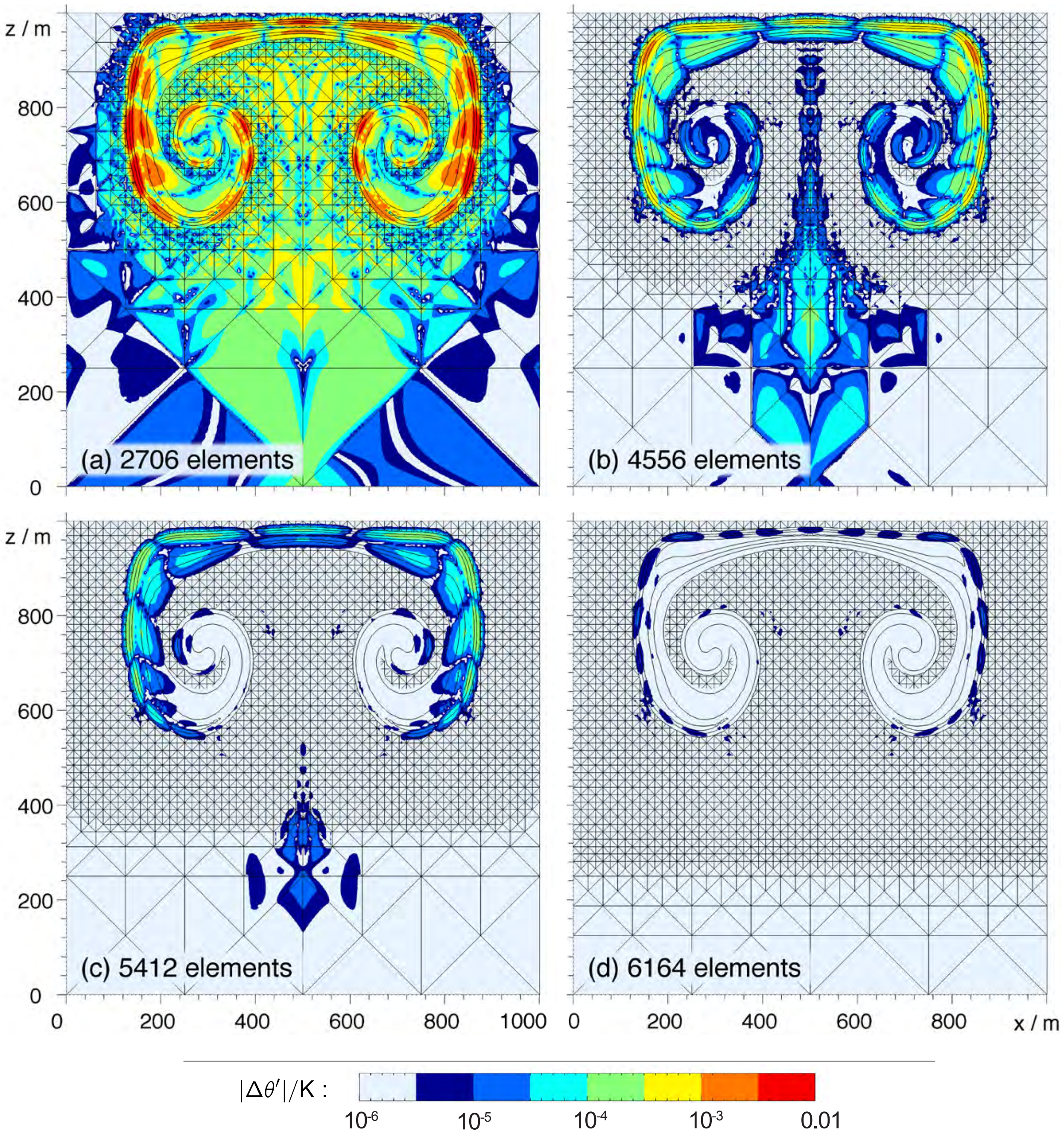

Figure 9: Comparison between adaptive and uniform simulations of a warm air bubble as in figure 8 at time $t = 800\,\mathrm{s}$. The same simulations as in figure 8 are shown. The labels of the different subfigures show the number of elements used at time $t = 800\,\mathrm{s}$. These numbers are different from those in figure 8 because the grid is continuously adapted to the current position of the warm air bubble.

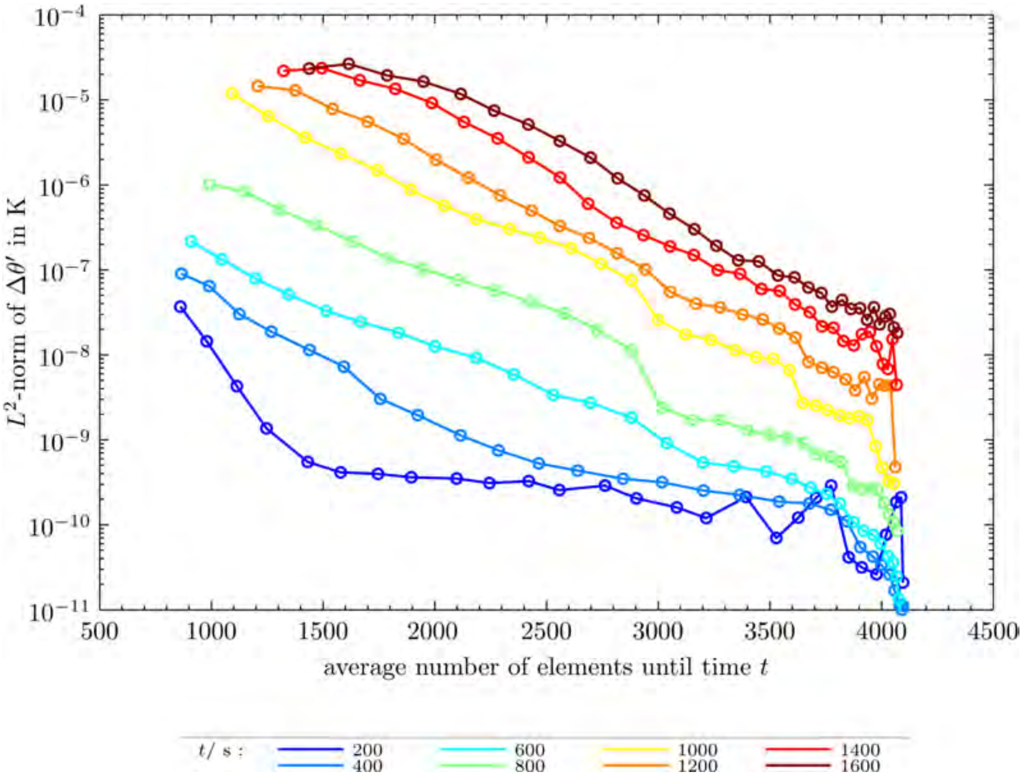

Figure 10: L^2-norm of the difference between adaptive and uniform simulations with a finest spatial resolution of $\Delta x_{\text{eff}} = 4.4\,\text{m}$. Each curve shows the result at a certain simulation time given in the legend. The data points on each of these curves belong to simulations with different sizes of the refinement region (similar to figure 8 and 9). The size of the refinement region is given by the average number of elements that were used until time t because this value is a good measure for CPU time that was used in these computations. The uniform simulation with $\Delta x_{\text{eff}} = 4.4\,\text{m}$ uses 4096 elements. For this reason this figure shows the complete transition from a fairly small refinement region towards the uniform simulation. For saving CPU-time only the left half of the warm air bubble was simulated for this figure. For this reason the number of elements in figures 8 and 9 is by a factor of two larger compared to this figure. As the results for the full domain are perfectly symmetric (see figure 8 and 9) we do not loose information by considering only the half of the domain in this figure.

5.2. Which Deviations Between Adaptive and Uniform Simulations are Significant?

As seen in the previous subsection adaptive simulations of our warm air bubble test case are not able to produce exactly the same results as in uniform simulations which use the finest resolution of the adaptive simulation. Therefore it is important to find a criterion to what extent the deviations between adaptive and uniform simulations are significant.

We can certainly neglect the difference if it is much smaller than the total numerical error of the uniform simulation. For comparing the numerical error between adaptive and uniform simulations we have to find some error indicator, because we do not have an analytical solution for the warm air bubble test case. Usually the error is estimated by computing a high-resolution reference simulation as in the paper of Straka et al. [33]. We wanted to avoid the additional computational cost of computing the reference simulation. For illustrating the error indicator that we are using figure 12 shows the left half of uniform simulations of the warm air bubble at time $t = 1000\,\text{s}$ for three different spatial resolutions. The results are very similar but the wave like perturbations in the boundary of the bubble at about $x \approx 200\,\text{m}$ and $z \approx 550\,\text{m}$ are quite different. Grabowski and Clark [12] investigate these perturbations in detail. A possible explanation for these perturbations is that numerical errors are amplified by instabilities that occur at the boundary of the warm air bubble due to a wind shear at the boundary of the bubble. Grabowski and Clark [12] investigate this behavior by adding predefined perturbations to the initial conditions. Their results demonstrate that the perturbations are growing exponentially in time until they finally form small vortices. At the same time

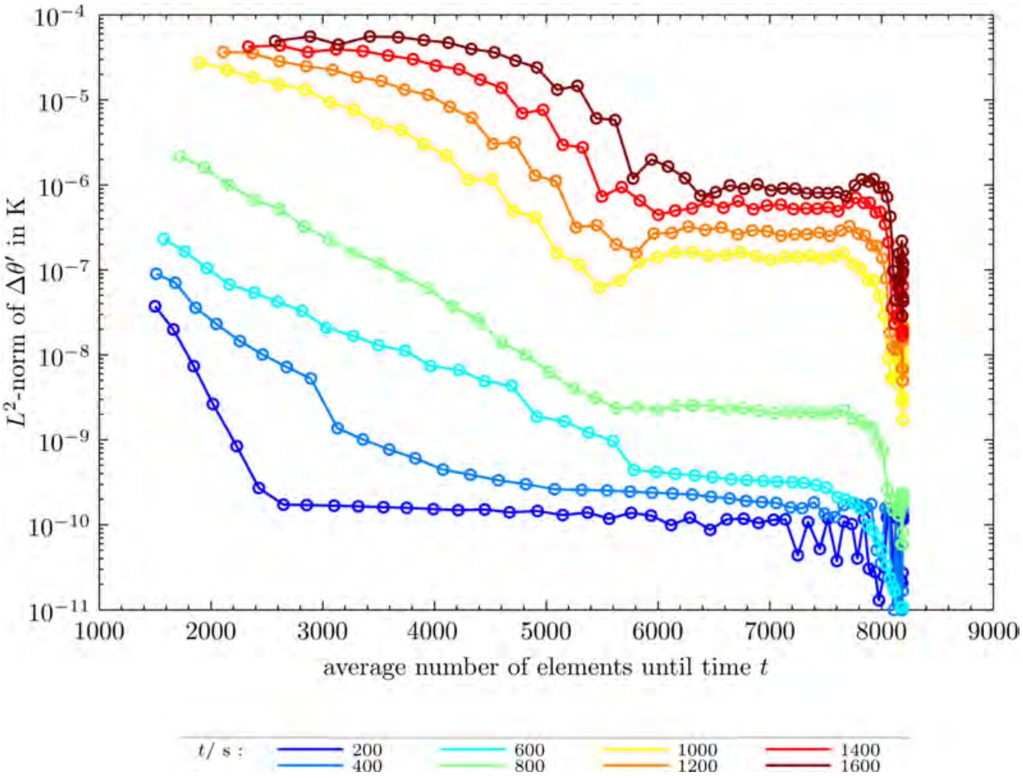

Figure 11: L^2-norm of the difference between adaptive and uniform simulations as in figure 10 with a finest spatial resolution of $\Delta x_{\text{eff}} = 3.1\,\text{m}$. The uniform simulation with $\Delta x_{\text{eff}} = 3.1\,\text{m}$ uses 8192 elements.

the spatial scale of these perturbations is growing. For this reason the resulting perturbations have a much larger spatial scale than the initial perturbations.

We use these perturbations in the shape of the warm air bubble at time $t = 1000\,\text{s}$ as a simple error indicator. For defining a quantitative error indicator figure 13 shows the left half of the warm air bubble for three different values of the viscosity parameter μ_{tc}. This figure demonstrates that the perturbations in the shape of the warm air bubble vanish with increasing viscosity. This behavior is reasonable because with increasing viscosity the correct result should become smoother and the numerical error should become smaller. Furthermore an increased amount of viscosity also reduces the instabilities at the boundary of the bubble by reducing the windshear. As the result with $\mu_{\text{tc}} = 0.2\text{m}^2/\text{s}$ does not show perturbations for $x > 180\,\text{m}$ we use the following definition for an error indicator d:

$$d := z_{\text{ref}} - z_{\text{test}} \tag{35}$$

where z_{ref} is the minimum height of the $0.15\,\text{K}$ contour for $180\,\text{m} < x < 500\,\text{m}$ and $z < 600\,\text{m}$ for the uniform simulation with $\mu_{\text{tc}} = 0.2\text{m}^2/\text{s}$ and $\Delta x_{\text{eff}} = 2.2\,\text{m}$ and z_{test} is the corresponding height in the simulation with $\mu_{\text{tc}} = 0.1\text{m}^2/\text{s}$ whose numerical error we want to estimate.

Figure 14 shows this error indicator d for 128 simulations with different sizes of the refinement region and three different finest spatial resolutions. As expected the error indicator is decreasing with increasing size of the refinement region as well as with increasing spatial resolution. Three examples of the simulations with $\Delta x_{\text{eff}} = 1.6\,\text{m}$ are shown in figure 15. When considering only the result for $x > 180\,\text{m}$ the results converge very nicely with increasing size of the refinement region towards the very smooth shape in figure 15c. However the perturbations for $x < 180\,\text{m}$ seem to be actually part of the correct solution. With increasing size of the refinement region they vanish when about 16000 elements are used at time $t = 1000\,\text{s}$ (figure 15b) but they come back when the size of the refinement region is further increased. For this reason

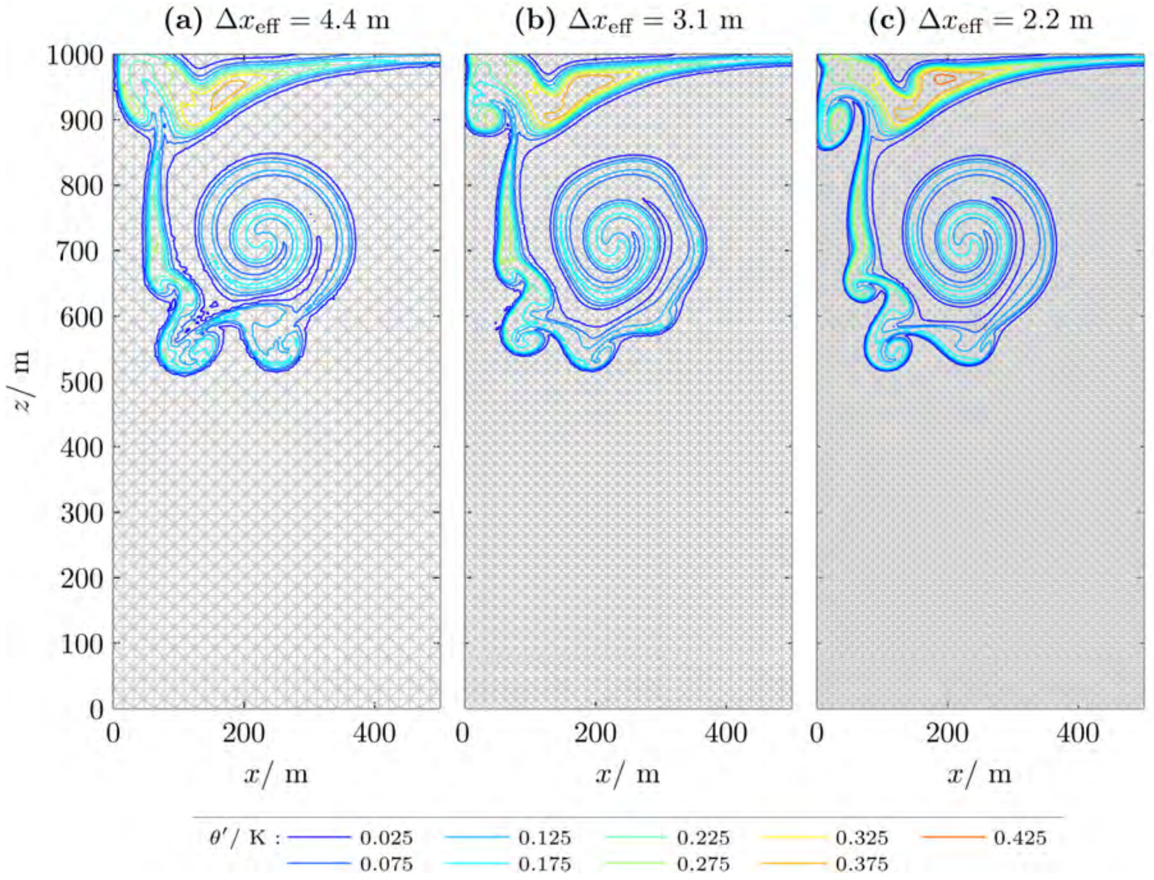

Figure 12: Uniform simulations of the left half of the warm air bubble from section 4.3 with a constant artificial viscosity of $\mu_{\text{tc}} = 0.1\,\text{m}^2/\text{s}$ at time $t = 1000\,\text{s}$ for three different spatial resolutions. Color contours show the deviation of potential temperature from the background state θ' from 0.025 K to 0.425 K with an interval of 0.05 K. The time-step of the IMEX-time-integration is $\Delta t = 0.01\,\text{s}$ in the case of the two coarser resolutions (a,b) and $\Delta t = 0.005\,\text{s}$ for the finest resolution (c).

we restrict our error indicator to $x > 180\,\text{m}$.

Figure 16 shows the error indicator d for the simulations of figure 14 as a function of the ratio between the number of elements of the adaptive simulation and the number of elements of the uniform simulation when using the same finest resolution. This figure indicates that for a very small refinement region the numerical error is (besides one outlier) almost independent of the spatial resolution when a certain ratio between the number of elements of the adaptive simulation and the number of elements of the uniform simulation is used. This is reasonable because in those simulations the numerical error is probably dominated by the error of using adaptive mesh refinement with a refinement region which is too small for covering all small scale features of the flow. The resolution independence stops when the adaptive simulation uses more than about 40% of the number of elements of the corresponding uniform simulation.

With the help of the error indicator d we can now consider the question, when the additional error by using adaptive mesh refinement is negligible. As a measure for the relative additional error of adaptive mesh refinement figure 17 shows $(d_{\text{adaptiv}} - d_{\text{uniform}})/d_{\text{uniform}}$ as a function of the ratio between the number of elements of the adaptive simulation and the number of elements of the uniform simulation when both use the same finest resolution. With d_{adaptiv} we denote the error indicator d for the adaptive simulation and with d_{uniform} we denote the error indicator d of the uniform simulation. This figure indicates that the relative additional error by using adaptive mesh refinement is smaller than 1% if the adaptive simulation with a finest resolution of 3.1 m uses more than 50% of the number of elements of the uniform simulation

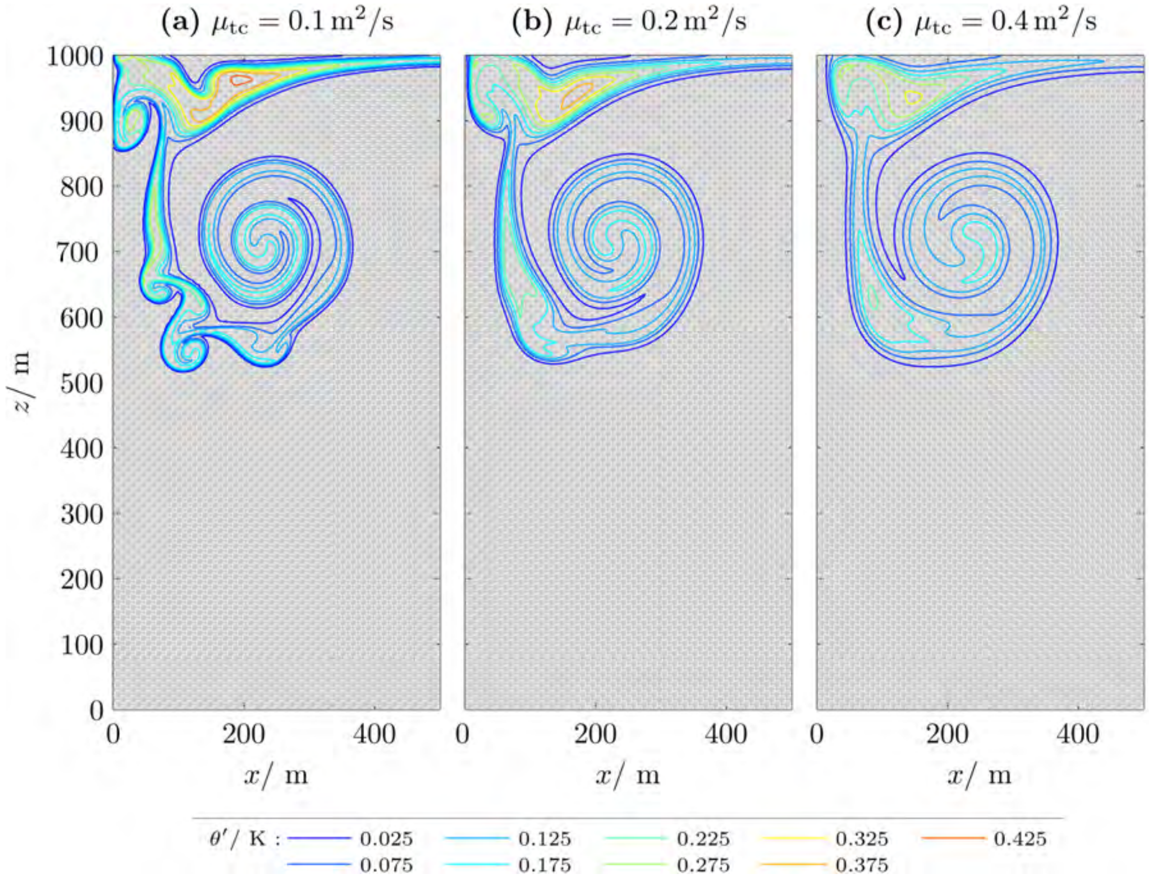

Figure 13: Uniform simulations of the left half of the warm air bubble from section 4.3 with a spatial resolution of $\Delta x_{\mathrm{eff}} = 2.2\,\mathrm{m}$ for three different values of the constant artificial viscosity μ_{tc}.

with a resolution of 3.1 m. Figure 18a shows the adaptive simulation with a finest spatial resolution of 3.1 m which uses on average 50% of the number of elements used in a uniform simulation with a resolution of 3.1 m (figure 18b) and table 4 shows some details of the simulations in figure 18. According to figure 17 the additional numerical error by using adaptive mesh refinement in figure 18a is about 1% of the total numerical error of the uniform simulation in figure 18b.

The CPU-time needed for the uniform simulation in figure 18b is almost twice as large as the CPU-time needed for the adaptive simulation in figure 18a. Furthermore less than 10% of the CPU-time used for the adaptive simulation is used for the adaptation of the grid. This adaptation time is currently mainly used for extending the refinement region which starts in each time step with criterion (24). We have not tried to optimize this extension of the refinement region. Therefore we expect that the CPU-time in our adaptive simulations can still be significantly reduced.

With increasing resolution figure 17 indicates that the size of the refinement region has to be increased. However it is difficult to study the resolution dependence with our setup in detail. When the spatial resolution is coarser than 3.1 m the limiter produces a significant amount of diffusion and reduces our error indicator d without being more accurate. When a finer resolution is used the error indicator d is almost vanishing (figure 14) which makes it very difficult to distinguish between the numerical error of the uniform simulation and the numerical error by using adaptive mesh refinement.

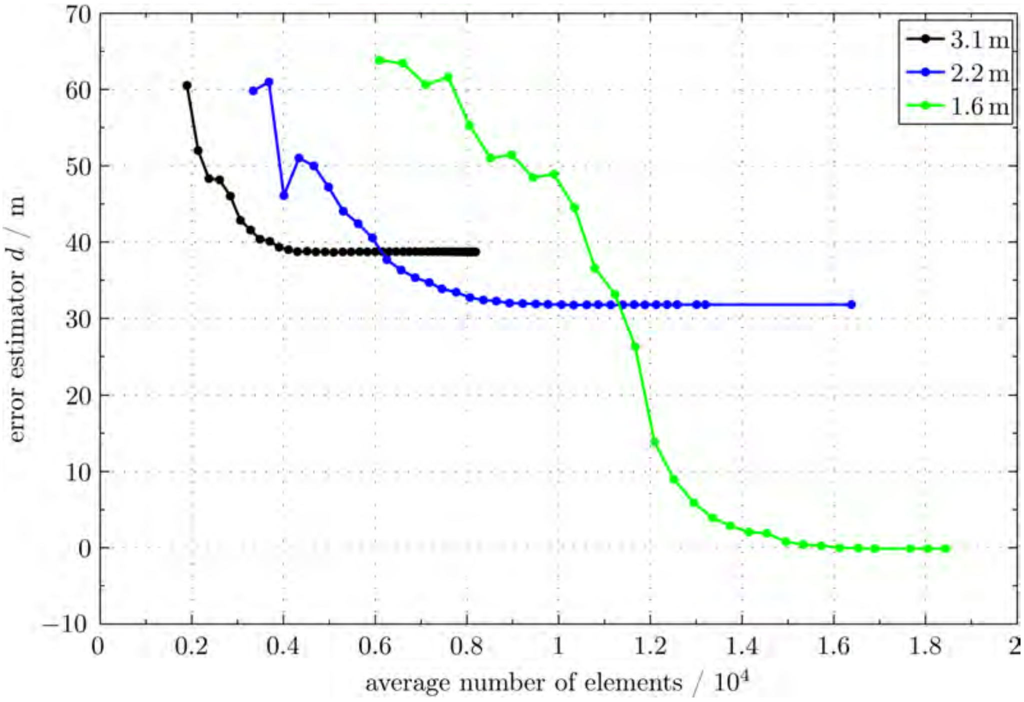

Figure 14: Error indicator d as a function of the average number of elements that were used in warm air bubble simulations until time $t = 1000\,\mathrm{s}$ for different sizes of the refinement region and three different finest spatial resolutions Δx_{eff} shown in the legend. Each data point in this figure represents a seperate simulation. As in figure 10 and 11 the size of the refinement region is given by the average number of elements used until time $t = 1000\,\mathrm{s}$. In total 128 simulations are shown. The data point at the right end of the black and the blue curve represents the uniform simulation.

	adaptive	uniform
Δt	0.010 s	0.010 s
Δx_{eff}	3.12 m	3.12 m
L	11.05 m	11.05 m
#elements	4106.21	8192.00
total time	41281.53 s	75011.17 s
grid time	3939.45 s	0.32 s
$\max(\theta')$	0.42 K	0.42 K
$\min(\theta')$	0.00 K	0.00 K

Table 4: Details as in table 1–3 for the two simulations shown in figure 18 with IMEX-time-integration. The time values are the time used for reaching $t = 1000\,\mathrm{s}$ of the warm air bubble test case and the number of elements is an average value over the whole time. Compared to figure 7d (table 3) the uniform simulation in figure 18 was faster because in this section 5 we simulated only the left half of the bubble. However here the simulations were done until time $t = 1000\,\mathrm{s}$ whereas in table 3 an end time of $t = 700\,\mathrm{s}$ was used.

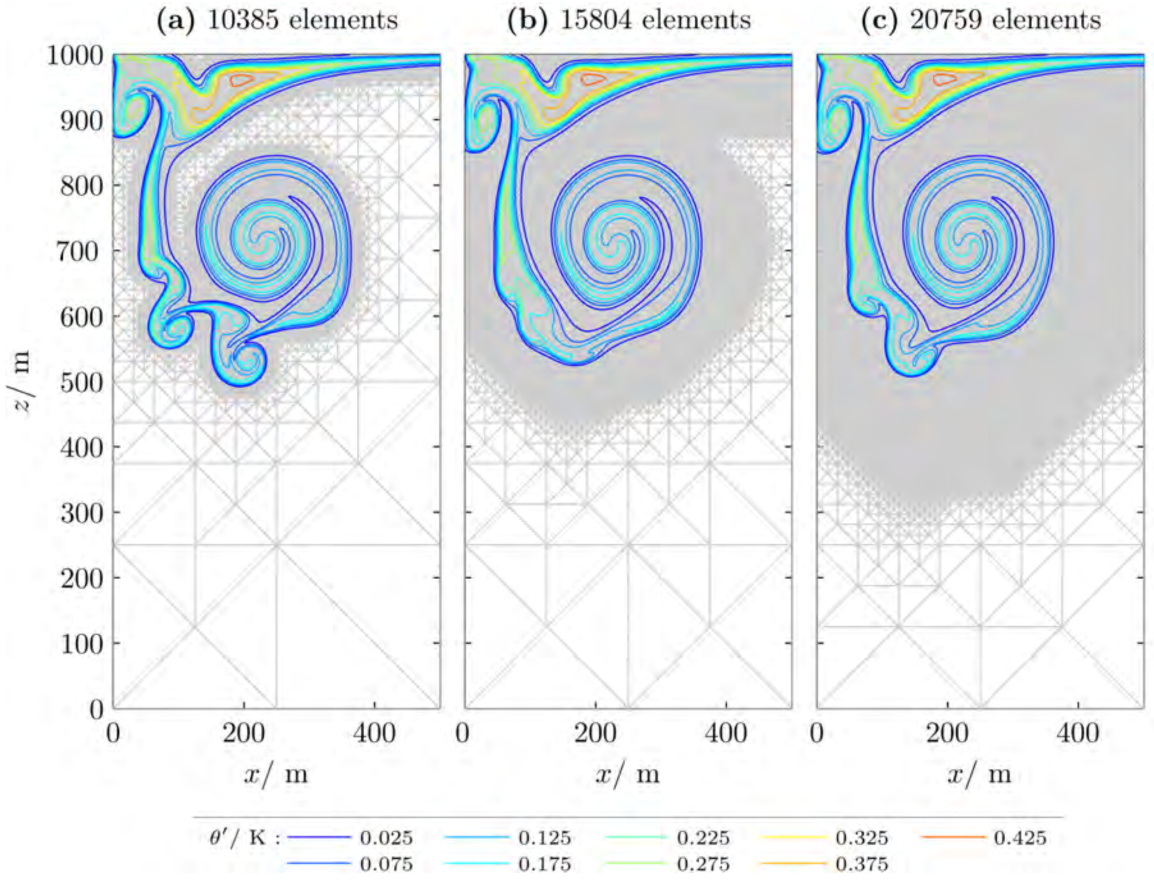

Figure 15: Left half of the warm air bubble at time $t = 1000\,\text{s}$ with a finest spatial resolution of $\Delta x_{\text{eff}} = 1.6\,\text{m}$ and three different sizes of the refinement region. The captions of the different subfigures show the number of elements at time $t = 1000\,\text{s}$.

5.3. Adaptive Versus Uniform for a Given CPU-Time

Except for the overhead needed for the adaptation of the grid (which is about 10% of the total CPU-time, see tables 1–4) the average number of elements used in our simulations is a good measure for the CPU-time needed for those simulations, because all our simulations in this section use the same polynomial order. For this reason figure 14 can easily be used for comparing adaptive and uniform simulations for a given CPU-time. The error indicator d of the uniform simulations with 3.1 m and 2.2 m is shown by the data point at the right end of the black and blue curve in figure 14. These values are both significantly larger than the error indicator for the higher resolved adaptive simulation which uses on average the same number of elements. Figure 16 indicates that the higher resolved adaptive simulations are more accurate than our uniform simulations when the adaptive simulations use more than about 40% of the number of elements of a high-resolution uniform simulation which corresponds to 80% of the number of elements of the coarser uniform simulation. As an example we consider the uniform simulation with $\Delta x_{\text{eff}} = 3.1\,\text{m}$: the adaptive simulation with a finest resolution of 2.2 m is more accurate when on average more than 40% of the number of elements of a uniform simulation with $\Delta x_{\text{eff}} = 2.2\,\text{m}$ are used. The uniform simulation with $\Delta x_{\text{eff}} = 2.2\,\text{m}$ uses twice the number of elements of the uniform simulation with $\Delta x_{\text{eff}} = 3.1\,\text{m}$. For this reason the adaptive simulation with a finest resolution of 2.2 m is more accurate than a uniform simulation with $\Delta x_{\text{eff}} = 3.1\,\text{m}$ when on average more than 80% of the number of elements of the uniform simulation with $\Delta x_{\text{eff}} = 3.1\,\text{m}$ are used.

In our results the adaptive simulation is more accurate than a uniform simulation which uses the same amount of CPU-time if the overhead by using adaptive mesh refinement is smaller than 20%.

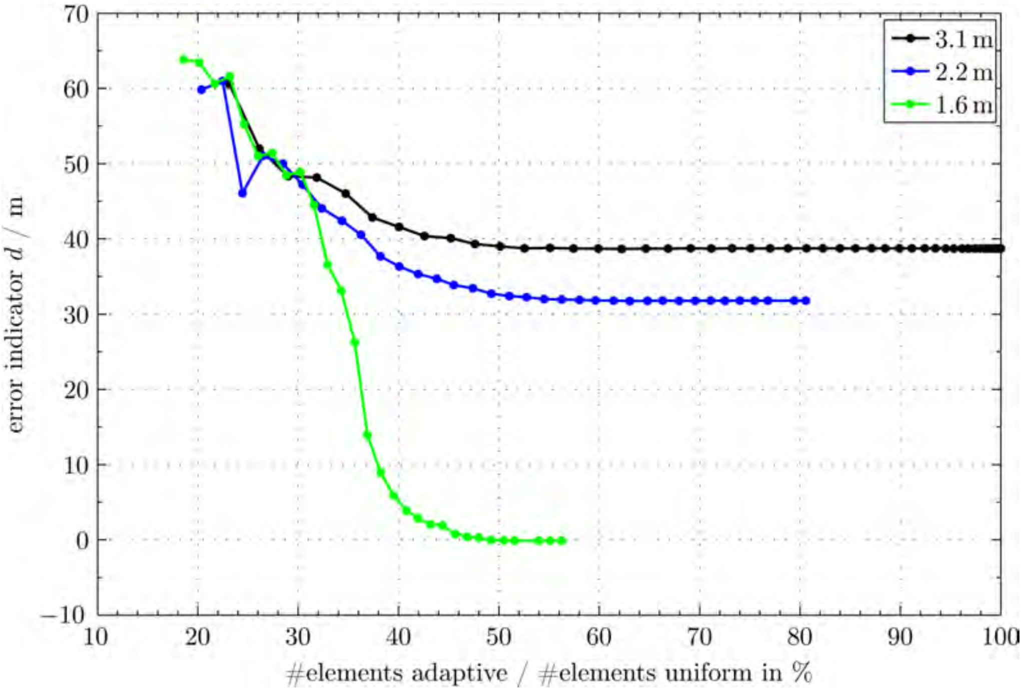

Figure 16: Error indicator d as a function of the ratio between the number of elements of the adaptive simulation and the number of elements of the uniform simulation in % when both use the same finest resolution. The legend shows the finest resolution Δx_{eff}.

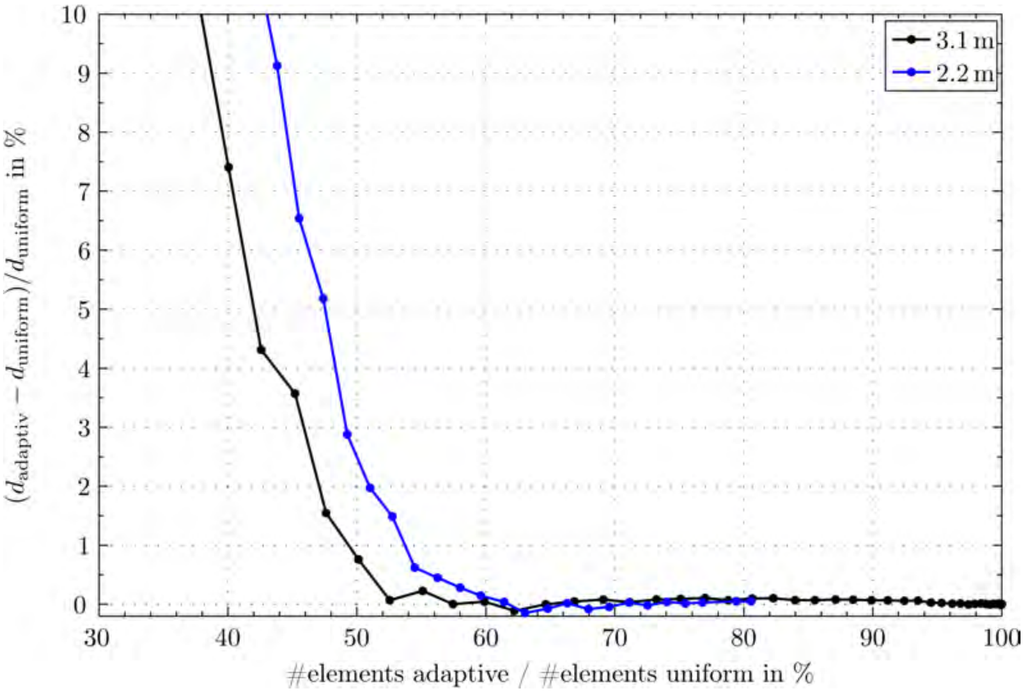

Figure 17: Relative additional error by using adaptive mesh refinement as a function of the ratio between the number of elements of the adaptive simulation and the number of elements of the uniform simulation in % when both use the same finest resolution. We use $(d_{\text{adaptiv}} - d_{\text{uniform}})/d_{\text{uniform}}$ for the relative additional error of adaptivity, where d_{adaptiv} is the error indicator d for the adaptive simulation and d_{uniform} is the error indicator d of the uniform simulation. Only values smaller than 10% are shown for making it easier to see details in the small values.

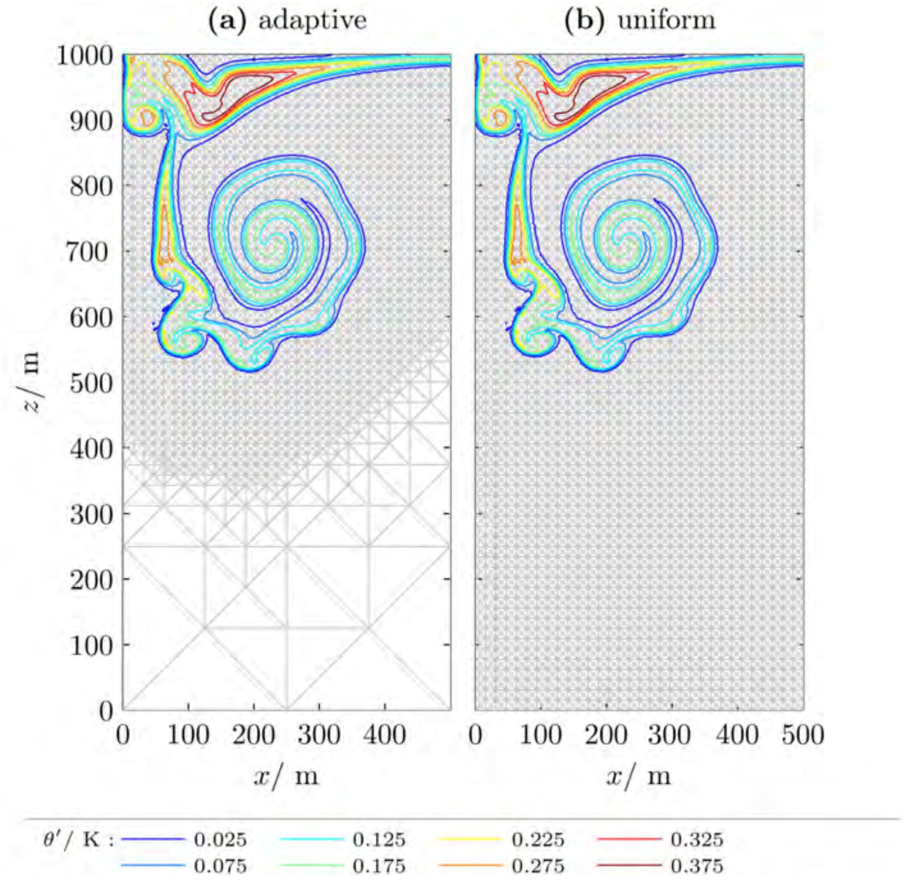

Figure 18: Direct comparison at time $t = 1000\,\mathrm{s}$ between the adaptive and uniform simulations for which the additional error by using adaptive mesh refinement is about 1% of the total numerical error of the uniform simulation (according to figure 17).

6. Summary and outlook

In this paper we present a novel numerical model for solving the 2D compressible Euler equations in Cartesian geometry. It uses a high-order discontinuous Galerkin method based on triangular elements [23] in combination with a IMEX-time-integrator [28]. This avoids the severe time-step restriction of explicit schemes. For the h-adaptivity we use the function library AMATOS [11] that uses a very efficient space filling curve approach. The choice of these numerical methods is motivated by the future application which the numerical model is designed for, namely the simulation of single cumulus clouds.

For testing our numerical model we simulate three standard test cases that are relevant for atmospheric convection. Our results agree well with the results from the literature. The adaptive mesh refinement uses a very simple refinement criterion: wherever the absolute value of the deviation of potential temperature from the background state exceeds a certain threshold a given finest resolution is used. This refinement region is extended by some fine elements to avoid small scale features moving into the coarse resolution. However we do not know a priori how large the refinement region should be. Therefore we investigate the influence of the size of the refinement region on a warm air bubble simulation. As a first step we consider the L^2-norm of the difference between an adaptive simulation and a uniform simulation which uses the finest resolution of the adaptive simulation. With increasing size of the refinement region the L^2-norm of the difference is continuously decreasing. The L^2-norm is getting quite small when a large refinement region is used. However we do not know how small the L^2-norm should be for being negligible.

The difference between adaptive and uniform simulations would certainly be negligible if the additional error by using adaptive mesh refinement is much smaller than the total numerical error of the uniform simulation. As the exact solution of our warm air bubble test case is not known we look for an approximate way to measure the accuracy of the simulation. We find that small perturbations in the shape of our warm air bubbles at time $t = 1000\,\mathrm{s}$ are strongly sensitive to numerical accuracy. For identifying these perturbations as clearly as possible we do not use any initial perturbations and we add artificial viscosity with a constant viscosity parameter. We define a simple error indicator d by measuring the size of those perturbations at the boundary of the warm air bubble.

This error indicator d allows us to estimate the relative additional error by using adaptive mesh refinement (figure 17). Our results indicate that the additional error by using adaptive mesh refinement for the warm air bubble test case is smaller than 1% of the total numerical error of a uniform simulation which uses the finest resolution of the adaptive simulation, if the adaptive simulation uses more than about 50% of the number of elements of the uniform simulation. Correspondingly the CPU-time needed for the adaptive simulation is up to a factor of two times smaller than for the uniform simulation. This comparison between adaptive and uniform simulations assumes that the uniform simulation uses the finest spatial resolution of the adaptive simulation for being able to resolve the same small scale features of the flow.

In many atmospheric applications the finest spatial scales of the flow cannot be resolved. Therefore it is often more important to compare the accuracy between adaptive and uniform simulations when both use the same amount of CPU-time. For this purpose we compare uniform simulations with a spatial resolution $\Delta x_{\mathrm{eff}}^{\mathrm{uniform}}$ with adaptive simulations with a finest spatial resolution of $\Delta x_{\mathrm{eff}}^{\mathrm{adaptive}} = \Delta x_{\mathrm{eff}}^{\mathrm{uniform}}/\sqrt{2}$. According to our error indicator d the adaptive simulations are more accurate than the uniform simulations when more than about 80% of the number of elements of the uniform simulations are used. Therefore the adaptive simulation of the warm air bubble test case is more accurate than a uniform simulation which uses the same amount of CPU-time if the overhead by using adaptive mesh refinement is smaller than 20% of the CPU-time of the uniform simulation. In our adaptive simulations the overhead by using adaptive mesh refinement was about 10%. However this overhead is dominated by our construction of the refinement region which is computed in each time-step. We expect that the overhead of adaptive mesh refinement can still be significantly reduced by optimizing the construction of the refinement region. This indicates that adaptive mesh refinement has the potential for improving the simulation of warm air bubbles.

It still remains an open question if this benefit exists in the same way for more realistic applications like the simulation of a single cumulus cloud. For the simulation of cumulus clouds we expect that the domain has to be very large for avoiding boundary effects. However it is not completely clear which resolution is necessary in the environment of the cloud. Probably it should be possible to use a fairly coarse resolution in the environment when modeling non-resolved turbulence with a sub-grid scale model (e.g., a Smagorinsky-model [42]). If this turns out to be true adaptive mesh refinement should have a big potential for future cloud simulations when the refinement criterion is carefully chosen.

To reach the ultimate goal of our project, that is to simulate a single cloud, requires the following further developments: first, we need to include moisture into our model. This makes it necessary to add variables for the water vapor content and the liquid water content of the air. As water vapor can condensate and liquid water can evaporate, both accompanied with a change of potential temperature, an additional heat source has to be added to eq. (3). We are working on solving the equations with these additional terms by implementing the approach used by Klemp and Wilhelmson [43]. Finally, we plan to extend this work to three-dimensions where we expect to modify our numerical models to handle either tetrahedral or hexahedral elements [34, 44].

We will also continue our work on estimating the error of adaptive numerical simulations. Questions that should be considered include: how does the size of a carefully chosen refinement region depend on the degree of the polynomials; how does our approach of using perturbations in the shape of warm air bubbles compare with different error indicators and is it possible to get similar efficiency results for moist air bubbles and for the extension to three dimensions?

7. Acknowledgements

We are grateful to the reviewers for giving many excellent suggestions for improving this paper. Financial support for this work was provided by the priority program MetStröm (SPP 1276) of the German Research Foundation (Deutsche Forschungsgemeinschaft). FXG gratefully acknowledges the support of the Office of Naval Research through program element PE-0602435N.

References

[1] W. Skamarock, J. Oliger, R. L. Street, Adaptive grid refinement for numerical weather prediction, J. Comput. Phys. 80 (1989) 27–60.

[2] G. S. Dietachmayer, K. K. Droegemeier, Application of continuous dynamic grid adaptation techniques to meteorological modeling. Part 1: Basic formulation and accuracy, Mon. Weather Rev. 120 (1992) 1675–1706.

[3] D. P. Bacon, N. N. Ahmad, Z. Boybeyi, T. J. Dunn, M. S. Hall, P. C. S. Lee, R. A. Sarma, M. D. Turner, K. T. Waight, S. H. Young, J. W. Zack, A dynamically adapting weather and dispersion model: the Operational Multiscale Environment Model with Grid Adaptivity (OMEGA), Mon. Weather Rev. 128 (2000) 2044–2076.

[4] N. N. Ahmad, D. P. Bacon, M. S. Hall, A. Sarma, Application of the multidimensional positive definite advection transport algorithm (MPDATA) to environmental modelling on adaptive unstructured grids, Int. J. Numer. Meth. Fl. 50 (2006) 1247–1268.

[5] R. Ford, C. C. Pain, M. D. Piggott, A. J. H. Goddard, C. R. E. de Oliveira, A. P. Umpleby, A nonhydrostatic finite-element model for three-dimensional stratified oceanic flows. Part I: Model formulation, Mon. Weather Rev. 132 (2004) 2816–2831.

[6] C. Jablonowski, R. C. Oehmke, Q. F. Stout, Block-structured adaptive meshes and reduced grids for atmospheric general circulation models, Philos. T. R. Soc. A 367 (2009) 4497–4522.

[7] H. Weller, T. Ringler, M. Piggott, N. Wood, Challenges facing adaptive mesh modeling of the atmosphere and ocean, B. Am. Meteorol. Soc. 91 (2010) 105–108.

[8] J. Behrens, Adaptive atmospheric modeling: key techniques in grid generation, data structures, and numerical operations with applications, Springer Verlag, 2006.

[9] T. Heus, H. J. J. Jonker, Subsiding shells around shallow cumulus clouds, J. Atmos. Sci. 65 (2008) 1003–1018.

[10] J. C. Wyngaard, Toward numerical modeling in the "terra incognita", J. Atmos. Sci. 61 (2004) 1816–1826.

[11] J. Behrens, N. Rakowsky, W. Hiller, D. Handorf, M. Lauter, J. Papke, K. Dethloff, amatos: Parallel adaptive mesh generator for atmospheric and oceanic simulation, Ocean Model. 10 (2005) 171–183.

[12] W. W. Grabowski, T. L. Clark, Cloud environment interface instability - rising thermal calculations in 2 spatial dimensions, J. Atmos. Sci. 48 (1991) 527–546.

[13] W. W. Grabowski, T. L. Clark, Cloud environment interface instability. 2. Extension to 3 spatial dimensions, J. Atmos. Sci. 50 (1993) 555–573.

[14] W. W. Grabowski, T. L. Clark, Cloud environment interface instability. 3. Direct influence of environmental shear, J. Atmos. Sci. 50 (1993) 3821–3828.

[15] D. A. Randall, Conditional instability of the 1st kind upside-down, J. Atmos. Sci. 37 (1980) 125–130.

[16] W. W. Grabowski, Cumulus entrainment, fine-scale mixing, and buoyancy reversal, Q. J. R. Meteor. Soc. 119 (1993) 935–956.

[17] J. P. Mellado, B. Stevens, H. Schmidt, N. Peters, Buoyancy reversal in cloud-top mixing layers, Q. J. R. Meteor. Soc. 135 (2009) 963–978.

[18] H. Johari, Mixing in thermals with and without buoyancy reversal, J. Atmos. Sci. 49 (1992) 1412–1426.

[19] W. W. Grabowski, Entrainment and mixing in buoyancy-reversing convection with applications to cloud-top entrainment instability, Q. J. R. Meteor. Soc. 121 (1995) 231–253.

[20] G. H. Bryan, J. C. Wyngaard, J. M. Fritsch, Resolution requirements for the simulation of deep moist convection, Mon. Weather Rev. 131 (2003) 2394–2416.

[21] R. Damiani, G. Vali, S. Haimov, The structure of thermals in cumulus from airborne dual-Doppler radar observations, J. Atmos. Sci. 63 (2006) 1432–1450.

[22] F. X. Giraldo, M. Restelli, A study of spectral element and discontinuous Galerkin methods for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling: Equation sets and test cases, J. Comput. Phys. 227 (2008) 3849–3877.

[23] F. X. Giraldo, T. Warburton, A high-order triangular discontinuous Galerkin oceanic shallow water model, Int. J. Numer. Meth. Fl. 56 (2008) 899–925.

[24] F. X. Giraldo, J. S. Hesthaven, T. Warburton, Nodal high-order discontinuous Galerkin methods for the spherical shallow water equations, J. Comput. Phys. 181 (2002) 499–525.

[25] M. A. Taylor, B. A. Wingate, R. E. Vincent, An algorithm for computing Fekete points in the triangle, SIAM J. Numer. Anal. 38 (2000) 1707–1720.

[26] S. Wandzura, H. Xiao, Symmetric quadrature rules on a triangle, Comput. Math. Appl. 45 (2003) 1829–1840.

[27] B. Cockburn, C. W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261.

[28] M. Restelli, F. X. Giraldo, A conservative discontinuous Galerkin semi-implicit formulation for the Navier-Stokes equations in nonhydrostatic mesoscale modeling, SIAM J. Sci. Comput. 31 (2009) 2231–2257.

[29] F. X. Giraldo, M. Restelli, High-order semi-implicit time-integrators for a triangular discontinuous Galerkin oceanic shallow water model, Int. J. Numer. Meth. Fl. 63 (2010) 1077–1102.

[30] M. Restelli, R. Sacco, L. Bonaventura, Semi–Lagrangian and semi–implicit discontinuous Galerkin methods for atmospheric modeling applications, Ph.D. thesis, Politecnico di Milano, 2007.

[31] F. X. Giraldo, M. Restelli, M. Lauter, Semi-implicit formulations of the Navier-Stokes equations: application to nonhydrostatic atmospheric modeling, SIAM J. Sci. Comput. 32 (2010) 3394–3425.

[32] L. E. Carr III, C. F. Borges, F. X. Giraldo, An Element-Based Spectrally Optimized Approximate Inverse Preconditioner for the Euler Equations, SIAM J. Sci. Comput. 34 (2012) 392–420.

[33] J. M. Straka, R. B. Wilhelmson, L. J. Wicker, J. R. Anderson, K. K. Droegemeier, Numerical-solutions of a nonlinear density-current - a benchmark solution and comparisons, Int. J. Numer. Meth. Fl. 17 (1993) 1–22.

[34] S. Gabersek, F. X. Giraldo, J. D. Doyle, Dry and moist idealized experiments with a two-dimensional spectral element model, Mon. Weather Rev. in press (2012) 46pp.

[35] J. VonNeumann, R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, J. Appl. Phys. 21 (1950) 232–237.

[36] P. O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods, AIAA paper 112 (2006) 2006.

[37] C. Altmann, A. Taube, G. Gassner, F. Lörcher, C. D. Munz, Shock detection and limiting strategies for high order discontinuous Galerkin schemes, in: K. Hannemann, F. Seiler (Eds.), Shock Waves, Springer Berlin Heidelberg, 2009, pp. 1053–1058.

[38] A. Kurganov, Y. Liu, New adaptive artificial viscosity method for hyperbolic systems of conservation laws, J. Comput. Phys. 231 (2012) 8114 – 8132.

[39] B. Cockburn, C. W. Shu, The local discontinuous Galerkin method for time-dependent convection-diffusion systems, SIAM J. Numer. Anal. 35 (1998) 2440–2463.

[40] K. Shahbazi, P. F. Fischer, C. R. Ethier, A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations, J. Comput. Phys. 222 (2007) 391–407.

[41] A. Robert, Bubble convection experiments with a semiimplicit formulation of the Euler equations, J. Atmos. Sci. 50 (1993) 1865–1873.

[42] J. Smagorinsky, General circulation experiments with the primitive equations, Mon. Weather Rev. 91 (1963) 99–164.

[43] J. B. Klemp, R. B. Wilhelmson, Simulation of 3-dimensional convective storm dynamics, J. Atmos. Sci. 35 (1978) 1070–1096.

[44] J. F. Kelly, F. X. Giraldo, Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: limited-area mode, J. Comput. Phys. 231 (2012) 7988 – 8008.